Guia de conversão de aplicação Simatic S7 para Logix5000

Solução de aplicação

Allen-Bradley • Rockwell Software

Rockwell Automation

Informações Importantes ao Usuário

Os equipamentos de estado sólido têm características operacionais que os diferenciam dos equipamentos eletromecânicos. As orientações de segurança para a aplicação, instalação e manutenção de controles de estado sólido (publicação SGI-1.1 disponível em seu representante de vendas Rockwell Automation local ou on-line, em http://literature.rockwellautomation.com) descrevem algumas diferenças importantes entre equipamentos de estado sólido e dispositivos eletromecânicos conectados por fios. Devido a esta diferença e também à ampla variedade de usos para equipamentos de estado sólido, todos os responsáveis pela aplicação deste equipamento devem estar seguros de que a aplicação pretendida deste equipamento é aceitável.

Em nenhuma circunstância, a Rockwell Automation, Inc. será responsável por danos indiretos ou resultantes do uso ou aplicação deste equipamento.

Os exemplos e gráficos contidos neste manual são apenas para fins ilustrativos. Devidos aos diversos requisitos e variáveis associados a todas as aplicações, a Rockwell Automation, Inc. não assumirá nenhuma responsabilidade pelo uso real baseado nos exemplos e gráficos.

Nenhuma responsabilidade sobre as patentes será assumida pela Rockwell Automation, Inc. em relação ao uso das informações, circuitos, equipamentos ou software descritos neste manual.

A reprodução do conteúdo deste manual, no todo ou em parte, sem permissão por escrito da Rockwell Automation, Inc., é proibido.

Ao longo deste manual, quando necessário, usamos observações para informá-lo das considerações de segurança.

ADVERTÊNCIA	Identifica informações sobre práticas ou circunstâncias que possam causar uma explosão em uma área classificada que leve a ferimentos ou morte, danos à propriedade ou perda econômica.
IMPORTANTE	Identifica informações críticas para a aplicação e compreensão bem-sucedidas do produto.
ATENÇÃO	Identifica informações sobre práticas ou circunstâncias que possam causar ferimentos ou morte, danos à propriedade ou perda econômica. As atenções ajudam você a identificar e evitar um perigo e a reconhecer a conseqüência.
PERIGO DE CHOQUE	Os rótulos podem estar sobre ou dentro do equipamento, por exemplo, um inversor ou motor, para alertar as pessoas de que pode haver uma tensão elétrica perigosa.
PERIGO DE QUEIMADURA	Os rótulos podem estar sobre ou dentro do equipamento, por exemplo, um inversor ou motor, para alertar as pessoas de que as superfícies podem atingir temperaturas perigosas.

Allen-Bradley, Rockwell Automation e TechConnect são marcas registradas da Rockwell Automation, Inc.

As marcas registradas não pertencentes à Rockwell Automation são de propriedade das empresas respectivas.

Sumário

Prefácio

Finalidade 7
Conversão *versus* tradução 7
Terminologia 8
Recursos adicionais 8
Serviços de conversão de lógica CLP fornecidos pela Rockwell Automation 9
- Recursos de serviços 9
- Serviços de conversão do programa CLP de interrupção única 9
- Benefícios do serviço 10
- Serviços oferecidos 10
- Pacote de conversão básico 10
- Pacote de limpeza inicial mais conversão 10
- Opções adicionais 11
- Conversões de programa adicionais disponíveis 11

Capítulo 1

Conversão de hardware

Introdução 13
Controladores S7 13
Sistemas de E/S 14
- E/S local S7 14
- Seleção e configuração de componentes de E/S do S7 14
- E/S local Logix 16
- Seleção e configuração de componentes E/S do Logix
- E/S remoto S7 20
- Configuração E/S remota de DP Profibus S7 21
- E/S distribuída Logix 22
- Configuração de E/S distribuída Logix 22

Redes 25
- Redes no S7 25
- Redes no Logix 27

Conversão de IHM 31
Conversão de sistemas contendo controladores distribuídos 32
- Implementação de hardware e software 32

Conectando dispositivos Siemens e Rockwell Automation 34
- Controladores 34
- Serviços distribuídos 34

Capítulo 2

Recursos do Logix que podem não ser familiares a usuários do S7

Introdução 35
Blocos de organização do S7 comparados a tarefas do Logix 36
- Blocos de organização no S7 36
- Tarefas no Logix 41
- Monitor de tarefas 46

Tags não endereçados 47
- Áreas de dados no S7 47
- Dados no Logix 50

E/S e tags de alias 51

Programando linguagens 53
Diagrama de lógica ladder do Logix 54
Texto estruturado do Logix 54
Diagrama de bloco de funções do Logix 55
Controle seqüencial de funções do Logix 55
Conversão de código STEP 7 para Logix 55
Vetores não indicadores 56
Instruções adicionais 57
Resumo de instruções adicionais 57
Tags backing 58
O Common Industrial Protocol (CIP) 58
Visualizando a rede 59
Intercâmbio de dados entre controladores 60
Envio/recebimento no STEP 7 60
Tags produzidos/consumidos no Logix 60
Tipos de dados definidos pelo usuário 61
Atualizando E/S assíncrona 62
O tipo de dados DINT 62
Gerenciador de fases 63
Gerenciamento de fases no STEP 7 63
Gerenciador de fases no Logix 63
Tempo de sistema (CST) 65
Entradas com data/hora 65
Saídas programadas 65
Sem variáveis temporárias 66
Sem a necessidade de acumuladores ou registradores especias 66

Capítulo 3

Conversão de sistema de software e funções padrão

Introdução 67
Funções de sistema Logix 68
Cópia 68
Ajuste e leitura de data e hora 69
Leitura de tempo do sistema 69
Manuseio de interrupções 70
Erros 70
Status – controlador 71
Status – Módulo 71
Status – para OBs e tarefas 72
Temporizadores 72
Rotinas de conversão 73
Rotinas de manuseio de grupo 73
Exemplos de chamadas de função do sistema 74
Ajustando o relógio 74
Desabilitando interrupções 76
Leitura de hora do sistema 78
Obter falhas 79
Informações do módulo 80
Obter tempo de varredura 81

Capítulo 4

Conversão de estruturas de programa típicas

Introdução 83
Conversão de exemplos de código 83
- Conversão de lógica ladder 83
- Saltos e tomada de decisões 90
- Vetores 94
- Tipos de dados do usuário 99
- Indicadores e vetores 102
- Máquina de estados 103
- Máquina de estados do STEP 7 104
- Grupos 108
- Variáveis temporárias do STEP 7 110
- Funções 110
- Cópia de bloco, COP e CPS 114
- Expressões matemáticas 116

Outros tópicos relacionados à programação 120
- Escopo de variáveis 120
- OBs, tarefas e seqüenciamento 120

Um exemplo maior - módulo de controle 121
- Componentes de CM 121
- Válvula de tipo de dados do usuário 122
- A instrução add-on 123
- Dados locais da instrução add-on 124
- Chamada 127

Capítulo 5

Erros comuns ao converter para Logix

Introdução 129
Não selecionar hardware apropriado 129
Subestimar o impacto do seqüenciamento de tarefas 130
Executar tradução em vez de conversão 130
Não usar as linguagens Logix mais apropriadas 130
Implementação de tipos de dados incorretos – DINT *versus* INT 131
- Adicionar DINTs 131
- Adicionar INTs 131
- Resultados de temporização 131

Código de usuário emulando instruções existentes 132
- Código do usuário 132
- Instrução COP 132

Uso incorreto de COP, MOV e CPS 133
Uso incorreto de CPT 133
Não manusear grupos de maneira ideal 133
Uso extensivo de saltos 133
Não usar tags com alias 133

Capítulo 6

Glossário S7 para Logix

Introdução 135
Terminologia de hardware 135
Terminologia de software 136

Apêndice A

Peças e equivalentes RA de S7 300 e S7 400

Introdução 139
CPUs compactas S7 300 140
CPUs padrão S7 300 140
CPUs com tecnologia S7 300 141
CPUs S7 300 livres de falhas 142
Módulos de entrada digital S7 300 142
Módulos de saída digital S7 300 143
Módulos de saída com relé S7 300 144
Módulos de combinação digital S7 300 144
Módulos de entrada analógica S7 300 144
Módulos de saída analógica S7 300 145
Módulos de combinação analógica S7 300 146
Módulos padrão S7 400 146
Controladores redundantes e livres de falhas 147
Módulos de entrada digital 147
Módulos de saída digital 147
Módulos de entrada analógica 148
Módulos de saída analógica 148

Apêndice B

Tabela de referência cruzada IHM Siemens

Micropainéis SIMATIC e equivalentes Rockwell Automation 149
Painéis SIMATIC - série 7x e equivalentes Rockwell Automation 151
Painéis SIMATIC - série 17x e equivalentes Rockwell Automation 152
Painéis SIMATIC - série 27x e equivalentes Rockwell Automation 155
Multipainéis SIMATIC - série 27x e equivalentes Rockwell Automation 157
Multipainéis SIMATIC - série 37x e equivalentes Rockwell Automation 159

Finalidade

Este manual do usuário oferece orientação para usuários e engenheiros que tenham usado os sistemas de controle com base em uma destas plataformas:

- Controlador S7 Siemens
- Controlador de automação programável Logix da Rockwell Automation (PAC)

E, além disso:

- tenham desejo ou necessidade de tirar proveito dos recursos PAC ou estejam nos estágios iniciais de migração do S7 para o Logix.
- tenham um código de programa STEP 7 específico que queiram converter para um código RSLogix 5000 eficaz e eficiente.

Use este manual para ajudá-lo a adotar boas práticas e para evitar erros comuns ao converter o projeto para Logix.

Conversão *versus* tradução

O tema de conversão *versus* tradução é repetido neste guia de conversão de aplicação. A simples tradução concentra-se apenas na linha de código e em encontrar um equivalente nas linguagens do Logix. Para converter uma aplicação de maneira ideal, você deve fazer mais do que apenas traduzir. Por exemplo, você pode se beneficiar de escolher uma linguagem de programação diferente, utilizando diferentes técnicas de programação e projetar um esquema de seqüenciamento diferente para resolver a mesma tarefa. Dessa maneira, a conversão é executada em um contexto de projeto de alto nível e de conhecimento dos pontos fortes do sistema Logix.

Se você tiver um código de aplicação a ser convertido, você deve compreender seu programa STEP 7 antes de iniciar a conversão – envolvendo-se em seu desenvolvimento ou lendo a documentação do programa e do processo que ele controla. Se o programa ou o processo não for familiar ou contar com pouca documentação, a conversão apropriada será difícil – será mera tradução com menos chances de sucesso. Por exemplo, no Logix, há um espaço de nome global, enquanto no ambiente Siemens há blocos de dados que podem ser carregados/descarregados pelo código da aplicação. Observar isso ajuda a projetar uma estratégia de conversão.

Em alguns casos, se a documentação tanto do processo quanto do programa não for completa, pode ser mais eficiente, em termos de duração do projeto/custo geral, escrever uma nova especificação e começar seu programa Logix com pouco tempo empregado na tradução do programa antigo.

Terminologia

O STEP 7 é o ambiente de software de programação dos controladores SIMATIC S7 da Siemens. O software RSLogix 5000 é usado com os controladores de automação programável Logix da Rockwell Automation. Nos referimos ao Logix como um controlador de automação programável devido a ele fazer muito mais do que um CLP para fins gerais tradicional. Ele fornece uma excelente plataforma de controle multidisciplina, um nomes de espaço comum, tempo de sistema para arquitetura multi-CPU verdadeiramente expansíveis, tipos de dados definidos pelo usuário e conectividade NetLinx total.

O termo "Logix" é usado para se referir a qualquer controlador ControlLogix, CompactLogix, GuardLogix, FlexLogix, DriveLogix ou SoftLogix, ou ao ambiente de programação RSLogix 5000 em que esteja claro o contexto ao qual estamos nos referindo.

Recursos adicionais

Cada seção deste guia de conversão de aplicação faz referência a outros manuais de usuário, guias de seleção e documentos da Rockwell Automation nos quais mais informações podem ser encontradas.

Número da publicação	Título da publicação
1756-SG001	Guia de seleção de controladores ControlLogix
1769-SG001	Guia de seleção de controladores CompactLogix 1769
1768-UM001	Manual do usuário de controladores CompactLogix 1768
1769-SG002	Guia de seleção de Compact I/O
1756-RM094	Manual de programação de considerações de projeto dos controladores Logix5000
1756-PM001	Manual de programação de procedimentos comuns dos controladores Logix5000
1756-RM003	Manual de referência de instruções gerais dos controladores Logix5000
1734-SG001	Guia de seleção de POINT I/O
1738-SG001	Guia de Seleção de ArmorPoint I/O
1792-SG001	Guia de seleção ArmorBlock I/O MaXum e ArmorBlock I/O
1794-SG002	Guia de seleção FLEX I/O e FLEX Ex
NETS-SG001	Guia de seleção NetLinx
VIEW-SG001	Guia de seleção de plataformas de visualização
IA-RM001	Arquitetura integrada: fundamentos da programação modular
6873-SG004	Diretório de produto do programa Encompass
1756-PM010	Manual de programação de instruções add-on dos controladores Logix5000
1756-RM087	Manual de referência de uso da memória e momento da execução dos controladores Logix5000
IASIMP-RM001	Manual de referência de literatura recomendada IA

Serviços de conversão de lógica CLP fornecidos pela Rockwell Automation

A Rockwell Automation fornece serviços adicionais para a conversão de lógica CLP.

- Recursos de serviço
- Serviços de conversão de programa CLP de interrupção única
- Benefícios do serviço
- Serviços oferecidos
- Pacote de conversão básica
- Pacote de limpeza inicial mais conversão
- Conversões de programa adicionais disponíveis

Recursos de serviço

Os serviços de conversão de programa converterão seu CLP herdado da marca Allen-Bradley ou programa de controlador programável de terceiros para execução em um sistema de controle de automação programável Logix ou os controladores programáveis SLC 500/MicroLogix ou PLC-5.

Os produtos herdados geralmente são mais caros de suportar e mais difíceis de consertar, o que pode aumentar o tempo parado e reduzir a produtividade. Por esse motivo, o suporte ao cliente Rockwell Automation agora oferece serviços de conversão de programa.Esses serviços são projetados par reduzir o custo e o tempo necessários para migrar de um CLP herdado par uma de nossas famílias de plataformas atuais PAC ou de controle de CLP.

Serviços de conversão de programa CLP de interrupção única

A migração para uma plataforma de controle Allen-Bradley atual a partir de um produto herdado aprimorará seu processo de produção, a confiabilidade e flexibilidade do sistema, oferecendo mais acesso ao poder de processamento do aplicativo e reduzindo os custos de consertos de equipamentos e estoque de peças sobressalentes. Com os serviços de conversão de programa do suporte ao cliente da Rockwell Automation, seu programa de controlador programável existente será convertido rápida e confiavelmente para a nova família de controladores. Os engenheiros do suporte ao cliente da Rockwell Automation podem ajudar na migração de equipamentos Allen-Bradley herdados para converter seus sistemas CLP para produtos da Rockwell Automation e, ao mesmo tempo, minimizando o tempo parado e maximizando o sucesso operacional.

Benefícios do serviço

Os especialistas de cada plataforma de produto estarão envolvidos durante o processo de conversão do programa. Não há anomalias difíceis de encontrar na lógica devido a erros de digitação. Na maioria dos casos, toda a tabela de dados é reproduzida e nenhum dado é perdido, bem como a documentação original é preservada, sem redigitação de comentários e símbolos. Os programas originais da marca Allen-Bradley podem estar no formato das séries 6200, APS ou AI. Os novos programas estarão no formato apropriado RSLogix.

Serviços oferecidos

Os pacotes de conversão do programa estão disponíveis, bem como pacotes personalizados específicos do projeto feitos caso a caso.

Pacote de conversão básica

- O programa de controlador programável será convertido no formato ControlLogix, CompactLogix, PLC-5 ou SLC 500/MicroLogix apropriado.
- O pacote fornece uma listagem de erros gerada durante a conversão que inclui instruções que não são convertidas diretamente e quaisquer endereços que não tenham sido convertidos, o que pode incluir indicadores e endereçamento indireto.
- O programa e a listagem de erros seriam retornados ao cliente para depuração e correção manual.

Pacote de limpeza inicial mais conversão

- O programa controlador programável original será convertido no formato ControlLogix, PLC-5 ou SLC 500/MicroLogix apropriado.
- Nós corrigiremos e converteremos qualquer instrução e/ou erros de endereçamento da família do novo processador.
- O programa concluído será, então, retornado ao cliente para partida e depuração.

Opções adicionais

As opções adicionais para os pacotes incluem o seguinte:

- Suporte por telefone para a aplicação durante a fase de partida e depuração do projeto.
- Consultoria sobre a reengenharia do sistema, interface de operação, estratégias de arquitetura e comunicação, para tirar proveito total das capacidades da nova plataforma que não sejam parte de um esforço de conversão de código, treinamento e partida no local estão disponíveis como um valor agregado em seu representante de vendas e soluções globais (GSS).
- Migração turn-key completa ou atualizações estão disponíveis em seu representante local GSS/sistemas com engenharia.

Conversões de programa adicionais disponíveis

- Formato PLC-2 para formato ControlLogix, CompactLogix, PLC-5, SLC500/MicroLogix
- Formato PLC-3 para formato ControlLogix, CompactLogix ou PLC-5
- Formato PLC-5/250 para formato ControlLogix ou CompactLogix
- Modicon – Quantum, 984, 584, 380, 381, 480, 485, 780, 785 para formato ControlLogix ou CompactLogix
- Siemens – S-5, S-7 para formato ControlLogix ou CompactLogix
- TI - 520, 520C, 525, 530, 530C, 535, 560, 560/565, 565, 560/560T, 560T, 545, 555, 575 para formato ControlLogix ou CompactLogix
- GE série 6 para formato ControlLogix ou CompactLogix

Conversões de programa de outros controladores programáveis de terceiros para programas de controlador Allen-Bradley também estão disponíveis. Entre em contato com o suporte técnico para obter mais detalhes.

Para programar um projeto de conversão ou aprender mais sobre os serviços de conversão de programa, entre em contato com seu representante de vendas Rockwell Automation local ou distribuidor autorizado: envie um e-mail para raprogramconversion@ra.rockwell.com, ou viste http://support.rockwellautomation.com/ e consulte a base de conhecimentos, documento G19154.

IMPORTANTE Use os serviços de consultoria para reengenharia, geralmente para expandir a funcionalidade do sistema e não para alterar o hardware por obsolescência ou motivos relacionados. As conversões de formato SLC para Logix e PLC-5 para Logix e geração de comentários PCE são criadas no software RSLogix 5000.

Observações:

Conversão de hardware

Introdução

O objetivo deste capítulo é oferecer orientação a um usuário ou engenheiro que precise determinar o hardware Logix correto como substituto do equipamento S7 existente.

O capítulo descreve como selecionar controladores, E/S local, E/S remota, redes e IHM, inclui uma seção sobre arquitetura de controladores distribuída e fornece exemplos de conversão de hardware dos módulos S7 mais freqüentemente usados.

Tópico	Página
Controladores S7	13
Sistemas de E/S	14
Redes	25
Conversão de IHM	31
Conversão de sistemas contendo controladores distribuídos	32
Conectando dispositivos Siemens e Rockwell Automation	34

Controladores S7

Esta tabela relaciona uma seleção relevante de controladores Siemens S7 usados em uma grande variedade de aplicações.

Seleção de controladores Siemens S7 atuais

Controlador	Nº da Peça	Equivalente Logix
313C	6ES7 313-5BF03-0AB0	L23 Serial
314C-DP	6ES7 314-6CG03-0AB0	L23 EtherNet/IP, L31
315-2 DP	6ES7 315-2AG10-0AB0	L32E, L32C
317-2 DP	6ES7 317-6TJ10-0AB0	L35CR, L35E
317T-2 DP	6ES7 317-6TJ10-0AB0	L43, L45
319-3 PN/DP	6ES7 318-3EL00-0AB0	L45, L61
414-2	6ES7 414-2XK05-0AB0	L61, L62
414-3 414-3 PN/DP	6ES7 414-3XM05-0AB0 6ES7 414-3EM05-0AB0	L62, L63, L64, L65

Seleção de controladores Siemens S7 atuais

315F-2 PN/DP (segurança)	6ES7 315-2FH13-0AB0 6ES7 317-2FK13-0AB0	GuardLogix L61S, L62S, L63S
414-H (Redundante) 417-H	6ES7 414-4HM14-0AB0 6ES7 417-4HT14-0AB0	L61-L65 com SRM
PCS7 – usa o controlador 417-4		L3x, L4x, L6x + FactoryTalk View, software FactoryTalk Batch

Um guia de adequação de alguns dos controladores S7 mais comumente utilizados:

- S7 315-2DP – Máquinas pequenas e médias.
- S7 317-2DP – Máquinas médias e grandes, aplicações de controle de processo pequenas e médias.
- S7 414-2 – Controle de máquina exigente, aplicações de controle de processo.
- S7 414-3 – Controle de máquina exigente, aplicações de controle de processo grandes.

O conjunto completo de controladores S7 está relacionado no Apêndice A.

Sistemas de E/S

Estas seções descrevem os sistemas de E/S Logix que substituem o equipamento S7 existente.

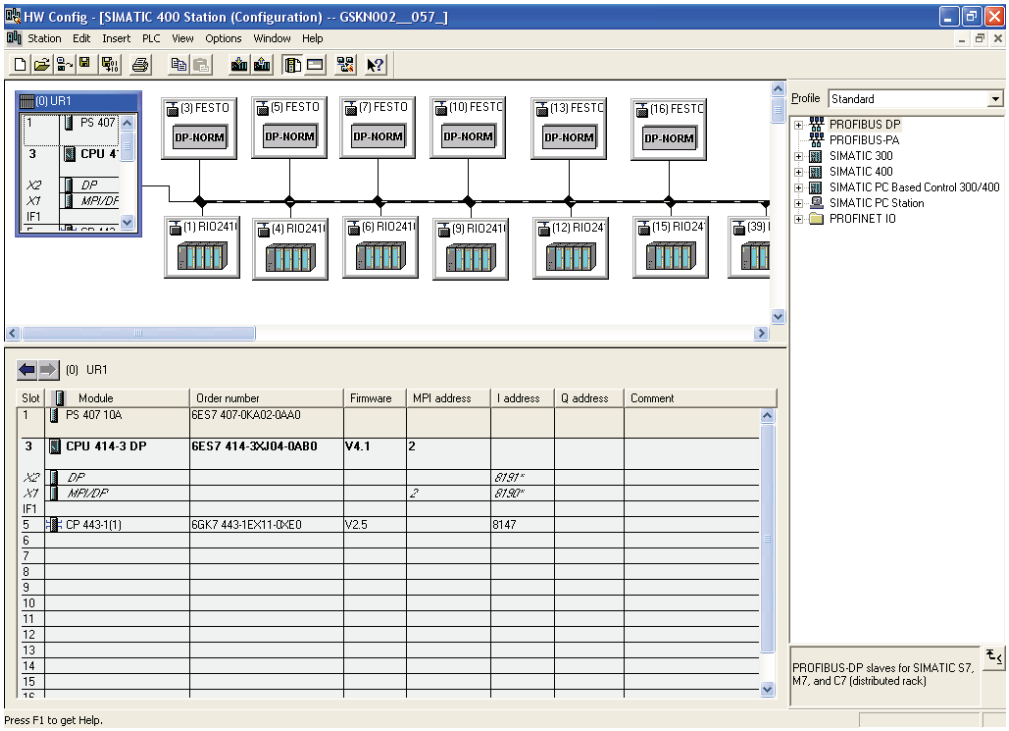
E/S Local S7

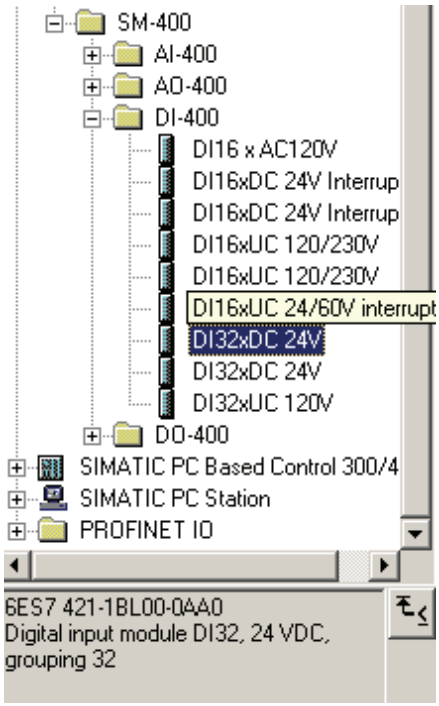
Há uma grande quantidade de módulos de E/S S7-300 e S7-400. Os módulos S7-300 são montados em um trilho DIN padrão e conectados a placas adjacentes usando conectores U, fornecidos com os módulos. Os módulos S7-400 são montados no rack S7-400.

Seleção e configuração de componentes de E/S S7

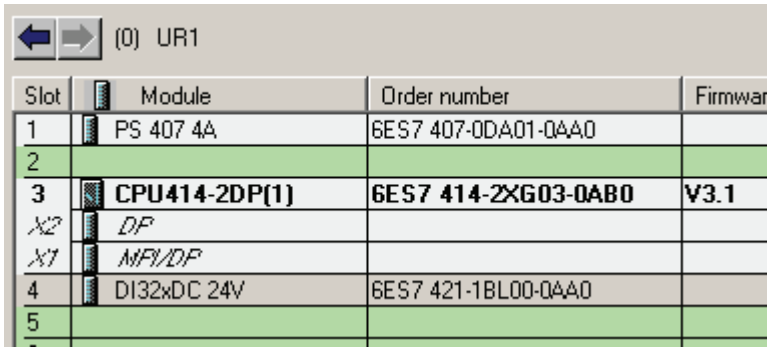
As capturas de tela a seguir são do programa de configuração de hardware STEP 7, um programa separado no conjunto de aplicações STEP 7.
No software RSLogix 5000, essa funcionalidade está totalmente integrada, como você verá posteriormente neste manual do usuário.

Programa de configuração de hardware STEP 7





Arraste o módulo selecionado para a tela do rack de configuração.



E/S Local Logix

Há disponível uma ampla variedade de módulos de E/S ControlLogix e CompactLogix. A E/S 1769 tem bom custo-benefício para as funcionalidades básicas freqüentemente solicitadas por OEMs, enquanto a família E/S 1756 oferece muitos recursos/funcionalidades para as aplicações mais exigentes, conforme solicitado pelos usuários finais e, muitas vezes, necessário para atender a níveis de desempenho específicos.

Os módulos CompactLogix são montados em um trilho DIN padrão e um sistema de acoplamento especial faz a conexão elétrica e mecânica com módulos adjacentes. Os engenheiros apreciarão o sistema de acoplamento mecânico — com o S7-300, os módulos são fixados apenas em um trilho especial e não uns aos outros (diferentemente do conector elétrico U).

Os módulos ControlLogix são montados em racks 1756.

- Para controladores 1769-L31, 1769-L32C, 1769-L32E e 1768-L43, o número máximo de módulos E/S conectados ao rack do controlador é 16, em até três bancos.
- Para controladores 1769-L35CR, 1769-L35E e 1768-L45, o número máximo de módulos de E/S conectados ao rack do controlador é 30, também em 3 bancos.
- Para controladores 1756, o número de slots no rack define o número máximo de módulos de E/S locais. Ele pode ser 4, 7, 10, 13 ou 17.

Em ambas as plataformas, podem ser inseridas E/S adicionais em redes CIP, nas quais as redes EtherNet/IP e ControlNet oferecem a integração de E/S mais perfeita e transparente.

Esta tabela relaciona os equivalentes Logix de alguns módulos de E/S S7 bastante utilizados.

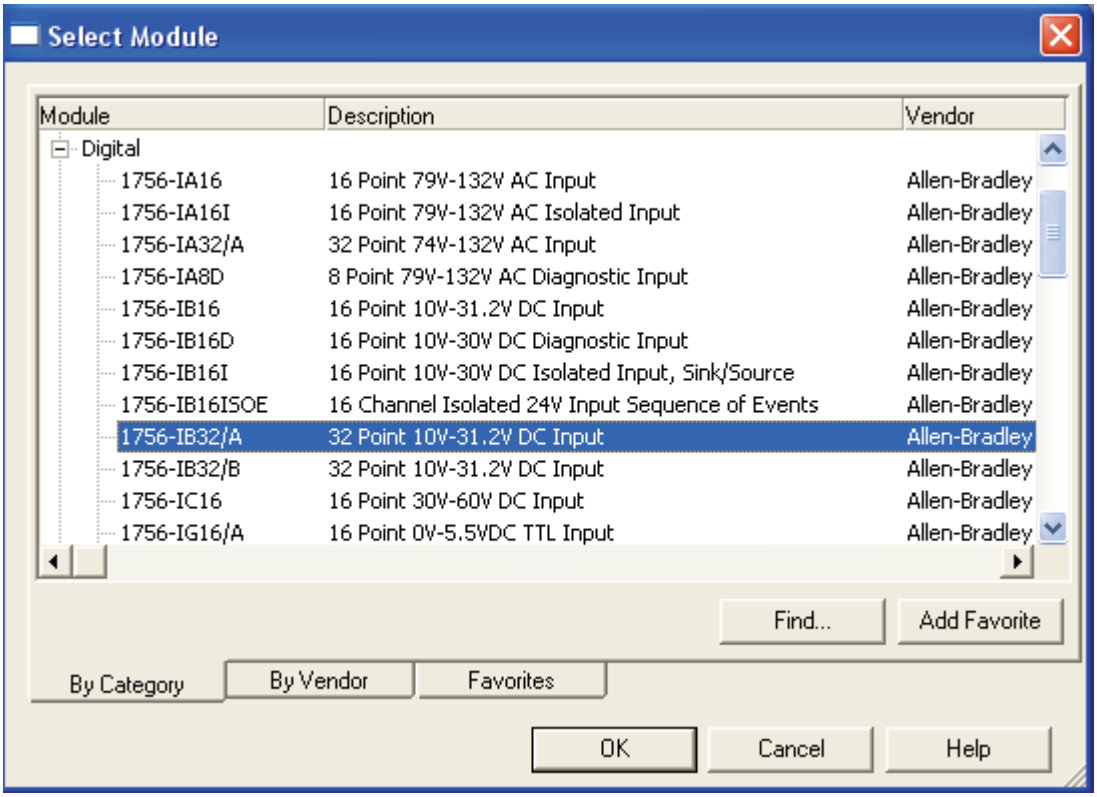
Equivalentes Logix de módulos de E/S S7

Módulo de E/S S7	Descrição	Equivalente Logix	Descrição
6ES7 321-1BL00-0AA0	S7-300 com entrada digital de 32 canais	1769-IQ32	CompactLogix com entrada digital de 32 canais
6ES7 322 - 1BH01-0AA0	S7-300 com saída digital de 16 canais	1769-OB16	CompactLogix com saída digital de 16 canais
6ES7 421-1BL01-0AA0	S7-400 com entrada digital de 32 canais	1756-IB32	ControlLogix com entrada digital de 32 canais
6ES7 422-1BH01-0AA0	S7-400 com saída digital de 16 canais	1756-OB16E	ControlLogix com saída digital de 16 canais

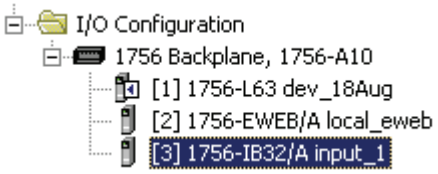
Consulte o Apêndice A para obter tabelas de conversão mais detalhadas de módulos de E/S.

Seleção e configuração de componentes de E/S Logix

A partir do ramo de configuração de E/S de sua árvore de projeto, a biblioteca Logix de perfis de dispositivo pode ser acessada. Esses perfis fornecem configuração totalmente dirigida por assistentes, para uma integração completa e fácil de usar na tabela de dados e controle programável intuitivo sobre cada funcionalidade do módulo, como conversão de escala, alarme e diagnósticos.



Selecione um item e ele aparecerá no rack em sua configuração de E/S.



Os tags do perfil de dispositivo do novo módulo de E/S foram adicionados automaticamente ao banco de dados de tags do escopo do controlador.

A visualização abaixo mostra os tags parcialmente expandidos.

⊟ Local:3:C			AB:1756_DI:C:0
⊞ Local:3:C.FilterOffOn_0_7			SINT
⊞ Local:3:C.FilterOnOff_0_7			SINT
⊞ Local:3:C.FilterOffOn_8_15			SINT
⊞ Local:3:C.FilterOnOff_8_15			SINT
⊞ Local:3:C.FilterOffOn_16_23			SINT
⊞ Local:3:C.FilterOnOff_16_23			SINT
⊞ Local:3:C.FilterOffOn_24_31			SINT
⊞ Local:3:C.FilterOnOff_24_31			SINT
⊞ Local:3:C.COSOnOffEn			DINT
⊞ Local:3:C.COSOffOnEn			DINT
⊞ Local:3:I			AB:1756_DI:I:0

O perfil contém dados de configuração e status, bem como dados de E/S.

⊟ Local:0:C			AB:1756_DI:C:0
⊞ Local:0:C.FilterOffOn_0_7			SINT
⊞ Local:0:C.FilterOnOff_0_7			SINT
⊞ Local:0:C.FilterOffOn_8_15			SINT
⊞ Local:0:C.FilterOnOff_8_15			SINT
⊞ Local:0:C.FilterOffOn_16_23			SINT
⊞ Local:0:C.FilterOnOff_16_23			SINT
⊞ Local:0:C.FilterOffOn_24_31			SINT
⊞ Local:0:C.FilterOnOff_24_31			SINT
⊞ Local:0:C.COSOnOffEn			DINT
⊞ Local:0:C.COSOffOnEn			DINT
⊟ Local:0:I			AB:1756_DI:I:0
⊞ Local:0:I.Fault			DINT
⊞ Local:0:I.Data			DINT

Consulte o Capítulo 4 para obter mais informações.

E/S remota S7

É comum dividir a E/S entre o rack local do controlador e as estações de E/S remotas, com a comunicação pela rede Profibus DP. Estes são os tipos de nós Profibus DP:

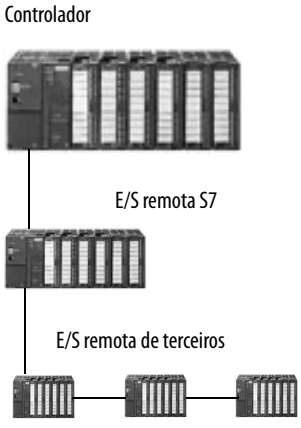
- A E/S remota S7, nesse caso os módulos de E/S S7-300 padrão, são montados em um painel de E/S remota e fazem interface com o barramento Profibus DP por meio de um módulo especial. O controlador enxerga essa E/S como a E/S local e atribui endereços de E/S padrão. Isso é chamado de ET200M.

- Outras E/S remotas Siemens, como ET200S (semelhante ao sistema POINT I/O) e ET200L (semelhante ao sistema FLEX I/O).

- E/S remotas de terceiros. Uma série de fabricantes de E/S e válvulas produz uma interface para vincular sistemas ao barramento Profibus DP da mesma maneira que a E/S remota S7. Para esses sistemas, um arquivo de integração especial (arquivo GSD) pode precisar ser importado para sua instalação de STEP 7.

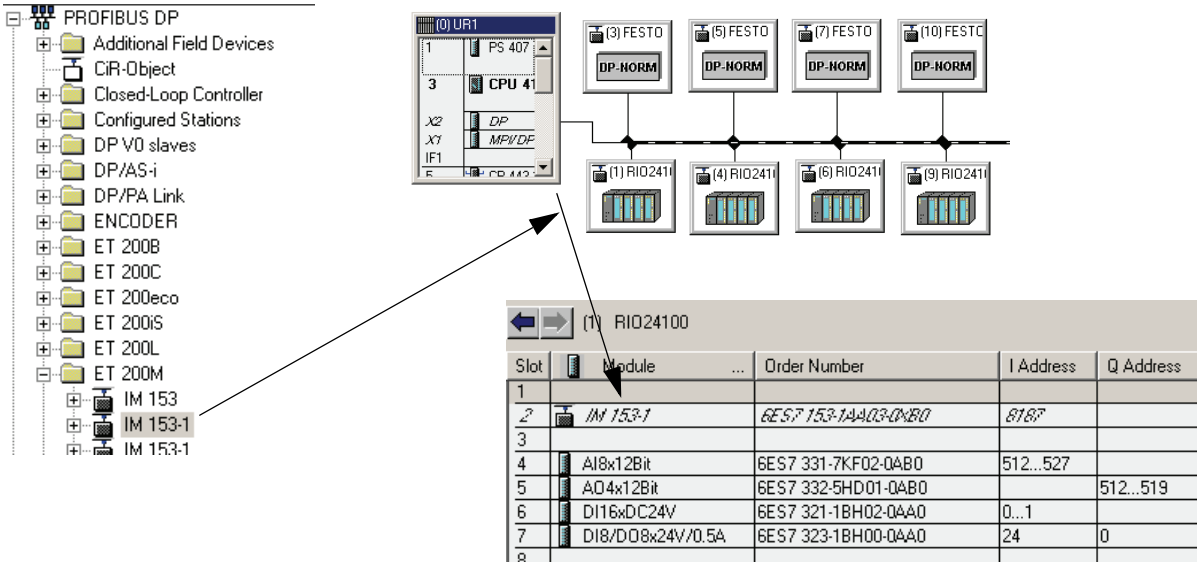
- Alguns fabricantes de dispositivos mais complexos, como balanças e inversores de velocidade variável , produzem interfaces Profibus DP para seus produtos.Para esses sistemas, um arquivo de integração especial (GSD) deverá ser importado para sua instalação de STEP 7. É necessário consultar freqüentemente a documentação do fabricante para aprender o significado das áreas de dados.

Configuração de E/S S7 típica



Configuração de E/S Remota de S7 Profibus DP

Um módulo de interface Profibus DP pode ser instalado na configuração de hardware sendo arrastado do catálogo de hardware para o gráfico do barramento Profibus DP. Após o módulo de interface ter sido instalado, ele poderá ser aberto e os módulos S7-300 padrão poderão ser adicionados como se fossem E/S locais.



A tabela de dados define os endereços E/S associados ao inversor. Os símbolos desses endereços devem ser adicionados manualmente na tabela de símbolos. A configuração de hardware está agora concluída.

É possível usar dispositivos remotos na rede Profibus DP juntamente com Logix, mas com as mesmas restrições/limitações de uso do ambiente S7.

E/S distribuída Logix

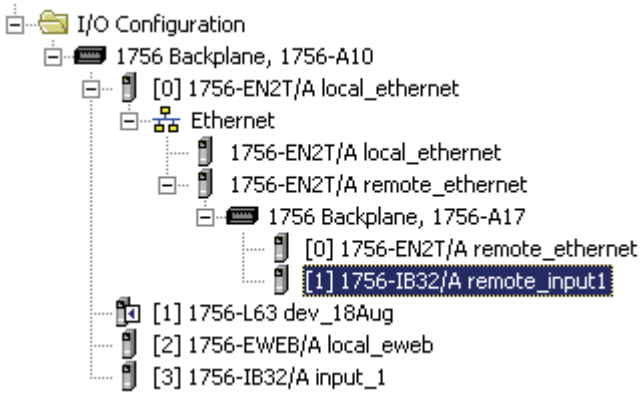
A E/S distribuída Rockwell Automation inclui E/S remota usando módulos de E/S 1756 ou 1769 e várias plataformas E/S distribuídas, como sistemas POINT I/O, FLEX I/O, ArmorPoint e ArmorBlock.

Os módulos E/S são conectados à rede usando um módulo de comunicação ou adaptador de comunicação ou diretamente por meio de uma interface de comunicação incorporada.

Configuração de E/S distribuída

Toda a configuração de E/S é feita na árvore de projeto do software RSLogix 5000. A partir da ramificação de configuração de E/S, insira um módulo de comunicação para seu tipo de rede selecionado.

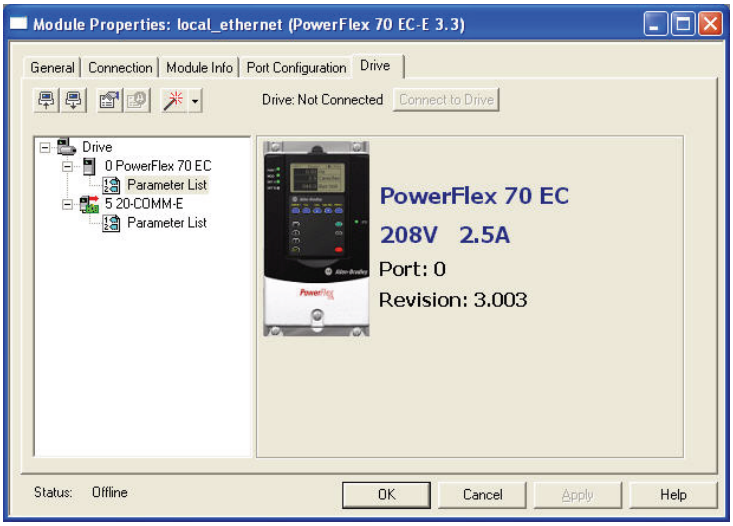
A captura de tela mostra a inclusão de um módulo de E/S remota 1756-IB32 conectado por meio de uma rede EtherNet/IP.



Observe que os tags correspondentes ao módulo de E/S remota foram adicionados automaticamente ao banco de dados de tag do escopo do controlador.

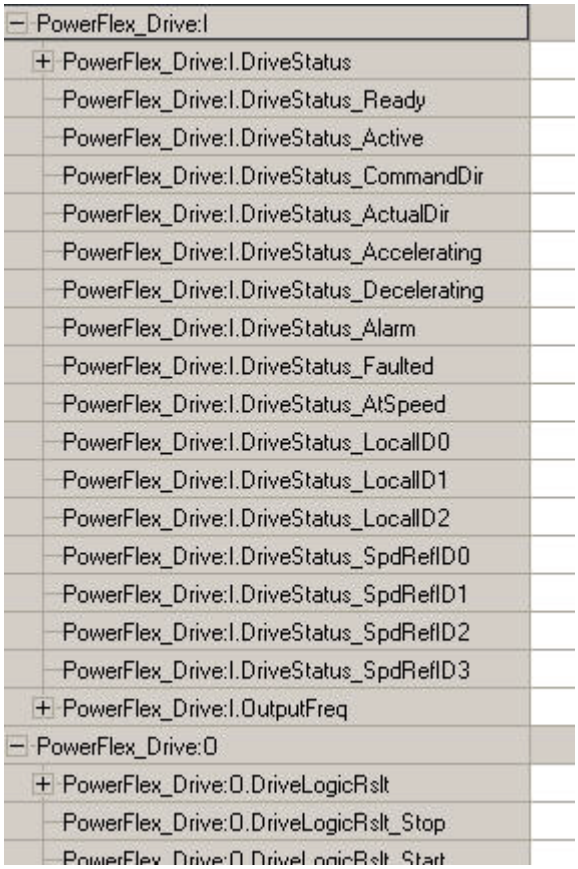
remote_ethernet:I			AB:1756_ENET_17SLOT:I:0
remote_ethernet:I.SlotStatusBits			DINT
remote_ethernet:I.Slot			AB:1756_ENET_SLOT:I:0[17]

Um inversor de velocidade variável ligado em rede, como o inversor PowerFlex, pode ser incluído da mesma maneira.



Novamente, o software RSLogix 5000 gerará novos tags automaticamente para qualquer dispositivo com um perfil no software RSLogix 5000 e conectado a uma rede EtherNet/IP ou ControlNet. Para a rede DeviceNet, a E/S GuardLogix Safety é integrada da mesma maneira. Outros dispositivos DeviceNet precisam ser configurados usando-se o software de configuração RSNetWorx e arquivos EDS que operam essencialmente de modo equivalente ao software gerenciador STEP 7 Profibus e arquivos GSD.

Mostrados abaixo estão os tags do perfil de dispositivo no software RSLogix 5000, disponível para centenas de dispositivos Rockwell Automation.



Redes

Consulte essas seções para obter informações sobre redes.

Redes em S7

Rede Profibus DP, DPV1, DPV3

No mundo S7, o principal tipo de rede para comunicação com dispositivos é a rede Profibus DP em várias implementações. Alguns controladores S7-300 de classe mais alta e todos os S7-400 têm portas mestre Profibus incorporadas.

Rede Profibus — outras

O FMS e o FDL Profibus são para comunicação de dados entre os controladores. Eles executam uma função semelhante à rede Ethernet industrial e a configuração é quase idêntica. As diferenças são que os processadores de comunicação Profibus são necessários, em vez da rede Ethernet, e que o cabeamento Profibus é usado.

O Profibus DPV2 pode ser usado para conectar servo-drives nos controladores S7-315T e S7-317T para controle de posicionamento de baixo nível.

Rede industrial Ethernet

A rede Ethernet industrial Siemens é uma variação da rede Ethernet em um ambiente industrial. Ela é usada principalmente para comunicação entre os controladores e para comunicação do controlador com o computador de programação.

A não ser por alguns controladores recentes equipados para Profinet, os controladores S7 não possuem portas Ethernet incorporadas. Um sistema S7 usando Ethernet industrial terá processadores de comunicação montados nos racks.

Dependendo do processador de comunicação, os seguintes protocolos podem ser usados:

- S7 (protocolo proprietário para comunicação entre controladores S7)
- TCP (Transmission Control Protocol) Raw Sockets
- ISO-on-TCP (Extended TCP com verificação adicional)
- UDP (User Datagram Protocol) Raw Sockets

O código da aplicação é necessário para gerenciar a maioria dos aspectos de comunicação dessas redes.

No ambiente Rockwell Automation, essa funcionalidade pode ser implementada usando as portas EtherNet/IP integradas, módulos EtherNet/IP Bridge e/ou módulos EWEB.

Profinet

Profinet fornece funcionalidade Profibus DP semelhante em uma Ethernet industrial com os mesmos requisitos de diretório de programação. Uma rede usando Profinet é semelhante a Profibus exceto pelo cabo e conectores diferentes e usa módulos de interface de campo Ethernet em vez de Profibus.
Os controladores com uma interface Profinet incorporada ou um processador de comunicação equipado para Profinet são usados para conexão com a rede.

Como alternativa, uma rede Profibus DP existente pode ser ligada em ponte com Profinet, com um proxy ou usando uma porta Profibus DP de um controlador equipado com Profinet.

Alguns módulos de interface de campo têm várias portas RJ45 com uma chave integrada para permitir um barramento de linha do tipo Profibus, se necessário.

A Profinet fornece estas três possibilidades de comunicação:

- Profinet CBA (Component Based Automation), que é usada principalmente para comunicação entre controladores e usa hardware Ethernet padrão e a pilha de software TCP/IP.
- Profinet IO para transferências programadas, como inversores ou módulos de E/S e usa hardware Ethernet padrão, mas ignora a pilha de software TCP/IP.
- Profinet IRT (Isochronous Real Time) para aplicações de controle de posicionamento que usa hardware Profinet específico e também ignora a pilha de software TCP/IP e deve existir em um segmento de rede protegido.

Se o framework Profinet CBA for usado, então as redes Profibus, Profinet e Ethernet industrial podem ser integradas pela configuração gráfica, com necessidade reduzida de programação adicional. As redes EtherNet/IP Rockwell Automation fornecem essa funcionalidade usando hardware padrão e a pilha de software TCP/IP padrão com funções incorporadas como a instrução Message (MSG) e tags produzidos/consumidos.

Redes no Logix

NetLinx é o termo que identifica a solução Rockwell Automation na área de tecnologias de rede. A seguir estão as principais redes usadas nos sistemas Logix:

- EtherNet/IP
- ControlNet
- DeviceNet

Essas redes têm vários recursos notáveis. Todas são projetadas sob o Common Industrial Protocol (CIP), que permite controlar, configurar e coletar dados sobre qualquer rede NetLinx. Como resultado, os dados podem fluir entre redes diferentes sem qualquer necessidade de software de conversão de protocolo ou proxies.

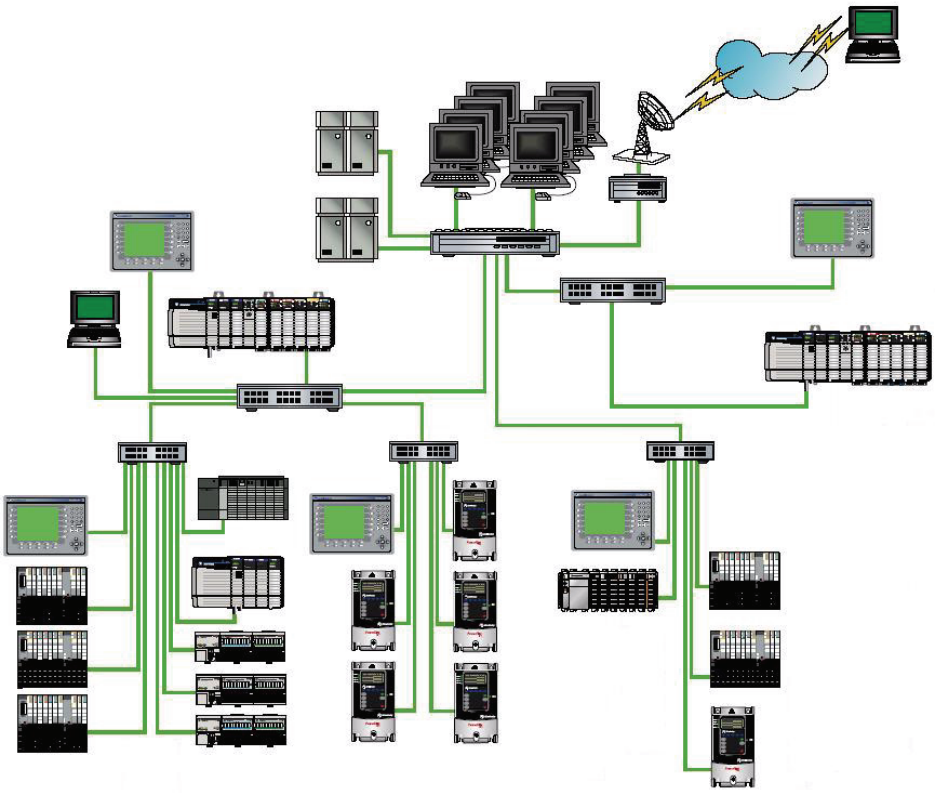
Um engenheiro que esteja se familiarizando com os sistemas Logix pode ficar impressionado pela natureza integrada e elegância na configuração de redes Logix.

Rede EtherNet/IP

A rede EtherNet/IP oferece uma suíte completa de serviços de aquisição de dados, configuração e controle. Ela usa TCP/IP para intercâmbio geral de mensagens/informações para mensagens de E/S. Ela é mais freqüentemente usada nestes tipos de configurações:

- Controle de E/S geral
- Intercâmbio de dados entre controladores
- Conexão de vários computadores
- Conexão de muitos dispositivos
- Conectividade com sistemas empresariais
- Integração de dispositivos de segurança
- Controle de posicionamento (futuro)

Exemplo de Ethernet/IP típico

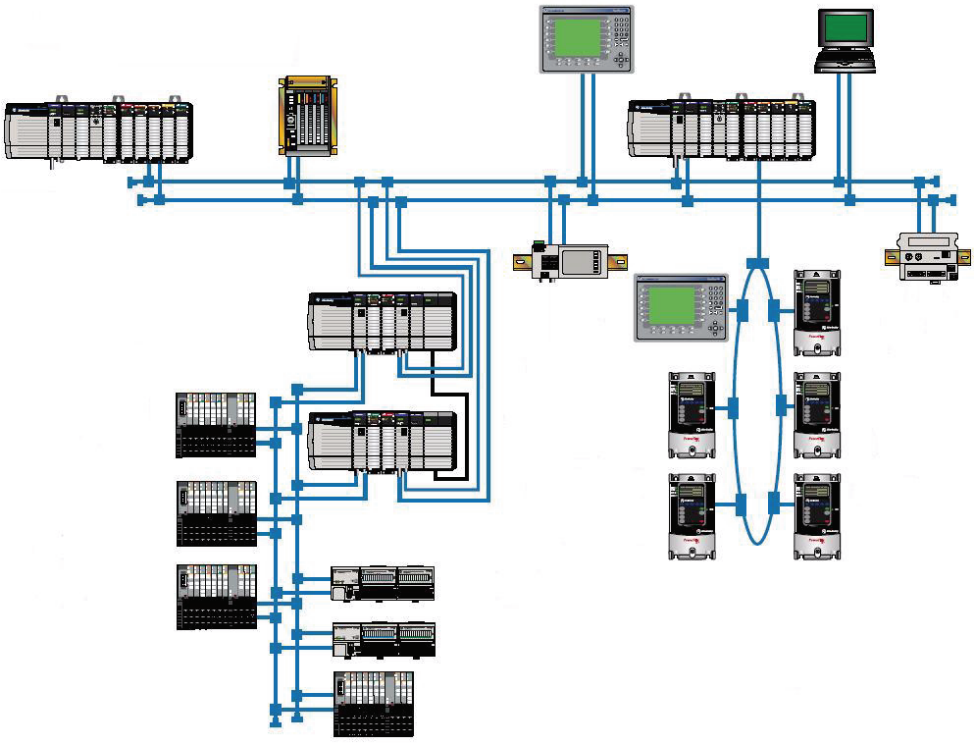


Rede ControlNet

ControlNet é uma rede de controle em tempo real que oferece transporte de alguns dados importantes de E/S, intertravamento e mensagem de dados, incluindo carregamento/descarregamento de dados de programação e configuração em um único link e mídia físicos. Ela é mais freqüentemente usada nestes tipos de configurações:

- Controle de E/S geral
- Intercâmbio de dados entre controladores
- Backbone para várias redes distribuídas DeviceNet

Exemplo de ControlNet típico

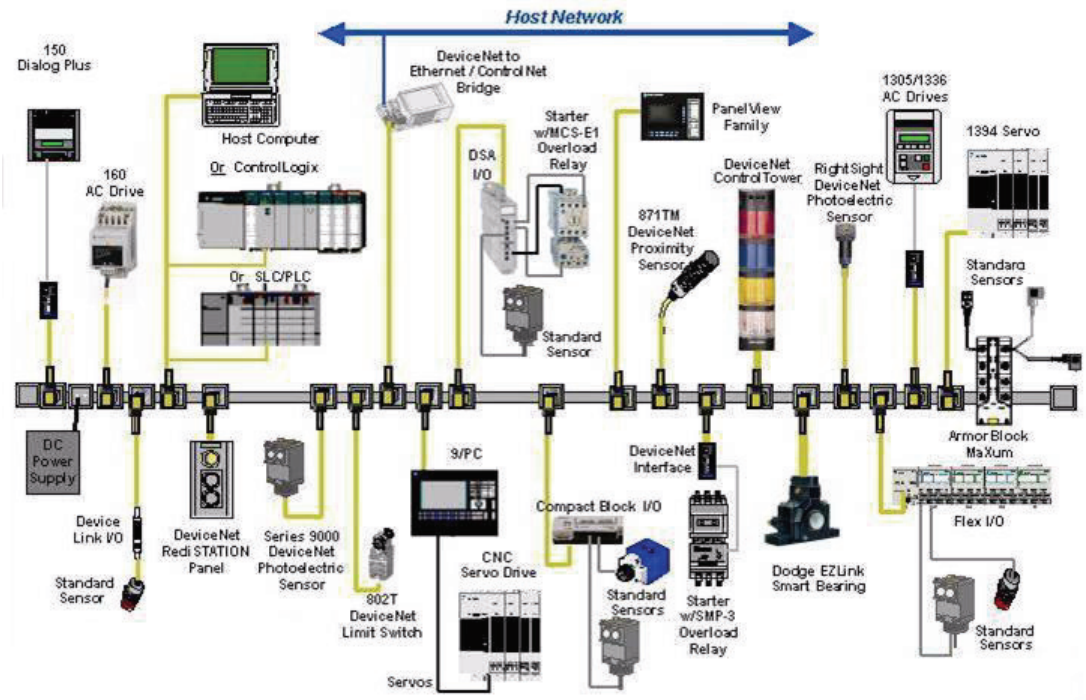


Rede DeviceNet

A rede DeviceNet é uma solução para redes de dispositivos industriais de baixo nível. Projetado para dispositivos com um baixo volume de dados por dispositivo para operação em tempo real. Ela é mais freqüentemente usada nestes tipos de configurações:

- Aplicações contendo dispositivos distribuídos com poucos pontos
- Rede de inversores de terceiros e outros dispositivos "simples" de terceiros
- Sistemas nos quais os dispositivos precisam estar conectados diretamente à rede com dados e energia na mesma conexão
- Quando as informações de diagnóstico avançado são necessárias

Exemplo de DeviceNet típico



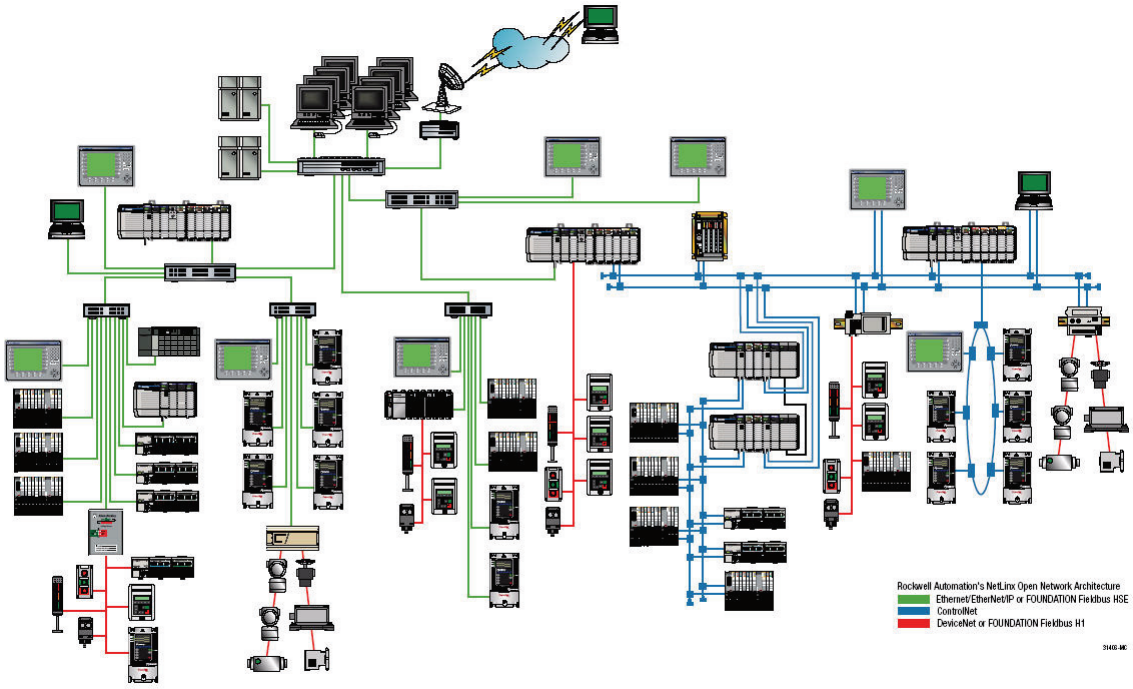
Interconectando redes NetLinx

Há dois caminhos comuns para interconectar redes NetLinx.

- Backplane de comunicação, permitindo vários links de rede de uma vez.
- Dispositivos de link de comunicação, vinculando duas redes juntas de maneira transparente.

Nenhum controlador ou programação é necessário nessas abordagens.

Exemplo de um sistema de controle em redes NetLinx



Conversão de IHM

Consulte o Apêndice B.

Conversão de sistemas contendo controladores distribuídos

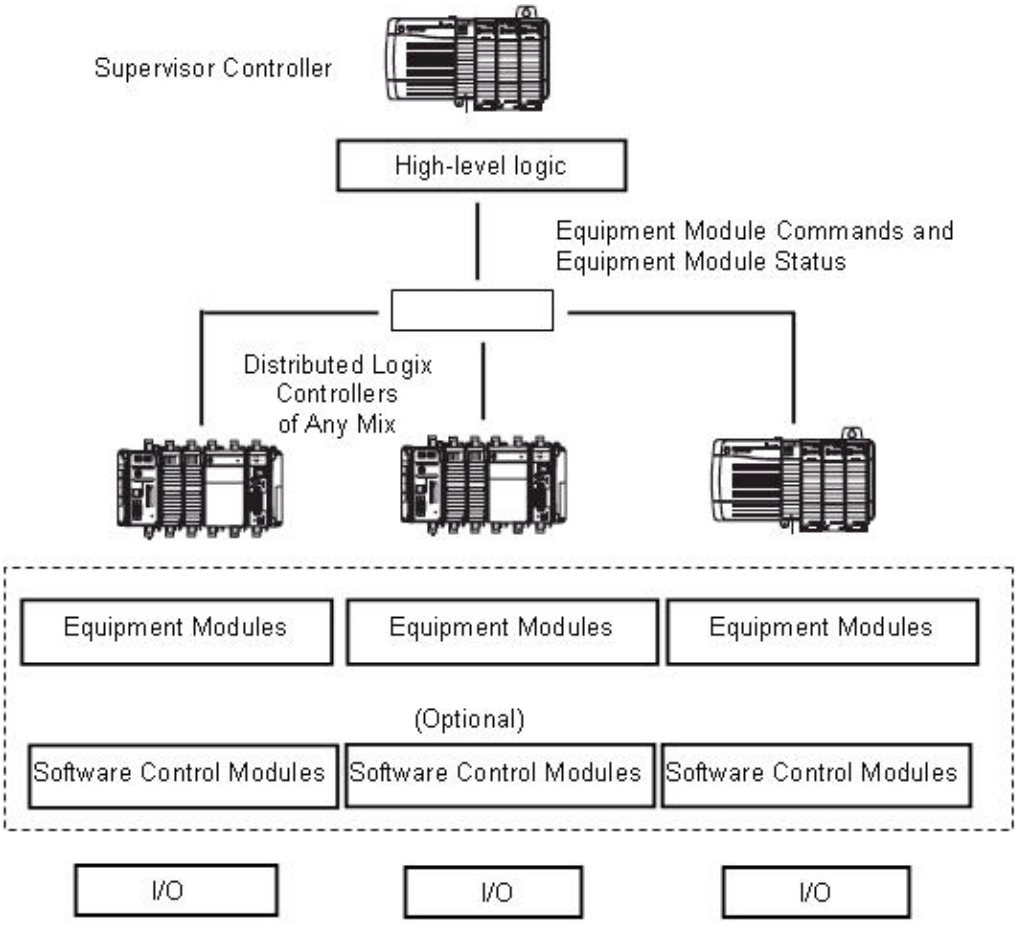
Esta seção abrange:

- como uma aplicação de controle discreta geral contendo um grupo de unidades funcionais pode ser criada usando vários controladores.
- como um método semelhante pode ser aplicado a uma aplicação de controle de processo projetada para o padrão S88.

Implementação de hardware e software

Controle discreto geral

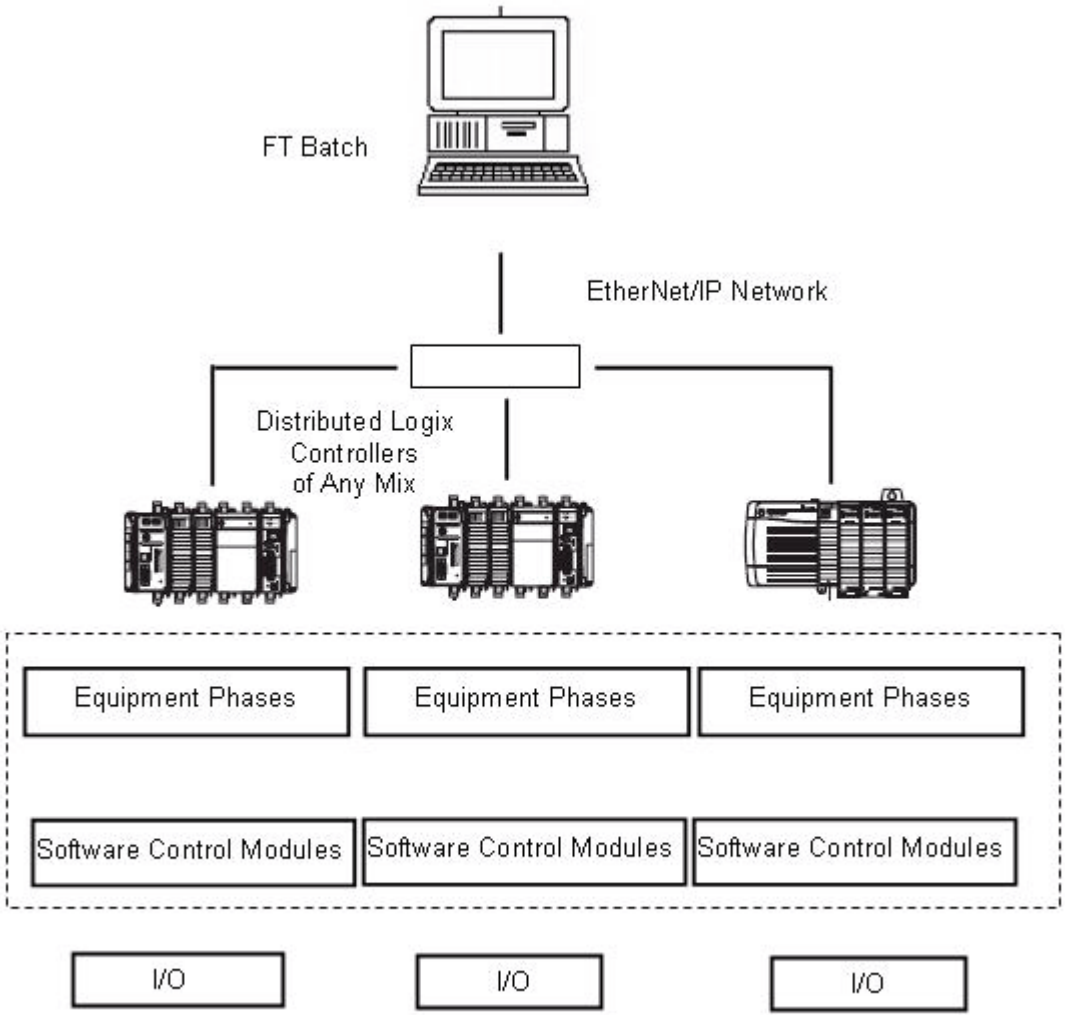
O modelo de hardware e software para lógica distribuída para controle discreto geral é mostrado abaixo. Nesse caso, a função de supervisão será desempenhada por um controlador. Uma rede EtherNet/IP ou ControlNet pode ser usada par interconectar os controladores. Produção/consumo ou mensagens explícitas podem ser usadas para intercambiar dados com o sistema.



Controle de processo

O diagrama abaixo ilustra a estrutura de hardware e software de uma aplicação de controle de processo S88. O PC executará o software FactoryTalk Batch, que é um pacote de software para conduzir a produção em batelada por meio de receitas. O software FactoryTalk Batch reside em um PC e comunica-se com cada controlador por meio da rede EtherNet/IP.

As fases do equipamento são configuradas sob o PhaseManager, conforme descrito posteriormente no Capítulo 2. Elas executam a lógica de fase e comunicam-se com a E/S do sistema de controle por meio de módulos de controle.



Conectando dispositivos Siemens e Rockwell Automation

Há situações em que você precisa interconectar equipamentos Siemens e Rockwell Automation. Recomendamos que você use produtos de empresas parceiras agrupados no programa Encompass.

Controladores

Os controladores Logix podem ser conectados a redes S7 usando:

- módulos em racks.
- conversores de protocolo de comunicação independentes.

Dispositivos distribuídos

Alguns dos sistemas de E/S da Rockwell Automation, inversores PowerFlex e terminais IHM conectam-se com Profibus por meio de adaptadores de comunicação, interfaces incorporadas ou módulos de interface.

Recursos Logix que podem não ser familiares a usuários S7

Introdução

Este capítulo descreve os recursos Logix que podem não ser familiares a usuários S7.

Tópico	Página
Blocos de organização do S7 comparados a tarefas do Logix	36
Tags não endereços	47
E/S e tags de alias	51
Linguagens de programação	53
Instruções add-on	57
O Common Industrial Protocol (CIP)	58
Intercâmbio de dados entre controladores	60
Tipos de dados definidos pelo usuário	61
Atualização de E/S assíncronas	62
O tipo de dados DINT	62
Gerenciador de fases	63
Tempo de sistema (CST)	65
Entradas com registro de data e hora	65
Saídas programadas	65
Sem variáveis temporárias	66
Sem necessidade de acumuladores ou registradores especiais	66

Determinados recursos do sistema Logix são mais fáceis de usar e manter que o S7 – por exemplo, os dados são organizados em bancos de dados de tags sem endereços absolutos, enquanto no S7 os itens de dados têm endereços absolutos selecionados pelo programador nas áreas de memória definidas.

Em outros aspectos, a estrutura do Logix é bastante semelhante ao S7, mas é apresentada de maneira diferente – por exemplo, debaixo da superfície, a estrutura tarefa é semelhante aos blocos de organização do S7.

Este capítulo contrasta os recursos diferentes (com os tags) e compara os recursos com similaridades subjacentes (como tarefas).

O objetivo é:

- permitir que o usuário S7 converta-se ao Logix com informações que tornarão o processo de design mais fácil e rápido.
- mostrar o que o Logix pode fazer, de modo que os engenheiros não tentem recriar o que existe no firmware do controlador.

Blocos de organização do S7 comparados a tarefas do Logix

Essa comparação de blocos de organização e tarefas apresentará a estrutura de um programa Logix a um usuário do S7.

Os blocos de organização e tarefas são semelhantes no fato de que ambos são chamados pelo sistema operacional do controlador, em vez de pelo programa do usuário. No STEP 7 (e Logix), há três tipos de blocos de organização (tarefa no Logix).

- O ciclo de programa OB (tarefa contínua no Logix), sendo que o OB recomeça do início após ser finalizado.
- A interrupção cíclica OB (tarefa periódica no Logix), sendo que o OB é executado em um período pré-configurado.
- A interrupção de hardware OB (tarefa de evento no Logix) é executada em resposta a alguns estímulos de hardware.

Muitos programadores de STEP 7 não usam a interrupção cíclica OB.

O Logix oferece um sistema operacional multitarefa configurável pelo usuário que permite que a energia da CPU seja alocada conforme a necessidade da aplicação.

Blocos de organização no S7

O tipo de OB é definido por seu número – eles são executados continuamente (OB1 apenas), executados periodicamente (OB30 – OB38), eles podem ser executados em eventos (OB40 – OB47) ou podem ser executados quando certas falhas aparecerem. Com o Logix, as tarefas não são numeradas, mas sim identificadas por um nome definido pelo usuário.

Um nome significativo pode ser anexado ao OB do STEP 7 se necessário.

Ciclo de programa OB1

O OB1 tem ciclo contínuo. Quando sua execução termina, os valores da tabela de imagem de saída são enviados às saídas, a tabela de imagem de entrada é atualizada com as saídas e o OB1 é novamente iniciado.

Um programa STEP 7 não tem que incluir o OB1, mas se este estiver incluído, será contínuo.

Fragmento de OB1 típico:

```
Network 3 : Title:
callup valve and motor control module

          CALL  "ValveMotor_Calls"

Network 4 : Title:
callup switch control module

          CALL  "Switch_Calls"

Network 5 : Title:
callup flow totalisers control module

          CALL  "Totaliser_calls"

Network 6 : Title:
callup analogue input control module

          CALL  "AnalogueIn_calls"
```

OB1 é a raiz da hierarquia de chamada de todo o código executado continuamente.

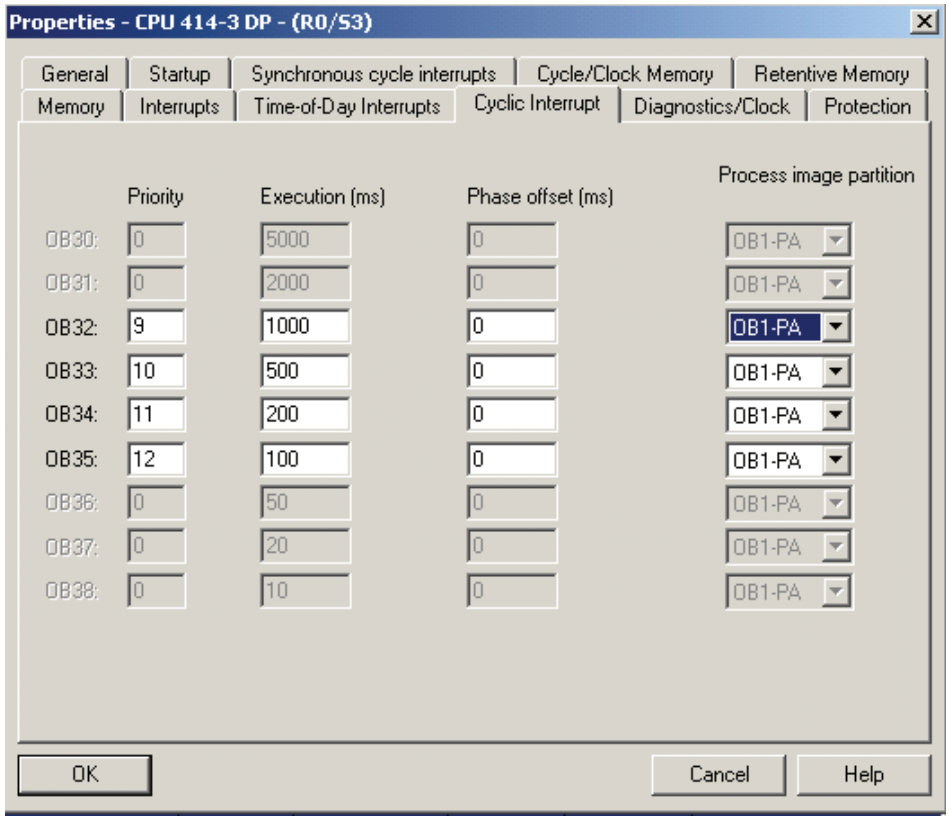
OB1 remonta a (apenas um possível, é claro) tarefa contínua no Logix. Na terminologia S7, OB1 é descrito como “ciclo de programa”.

Para os leitores que estejam mais familiarizados com Logix do que com STEP 7, pode ser útil saber que na lógica ladder do STEP 7, uma rede é o mesmo que uma linha Logix. Na lista de declarações do STEP 7, as redes ainda estão lá, mas servem apenas para aprimorar a aparência do código. Elas dividem o código em seções e permitem que componentes sejam adicionados. Todo o código deve ser colocado em uma rede, se desejado — ele deve ser compilado e executado perfeitamente bem.

OB30 – interrupções cíclicas OB38

Esses OBs são executados em intervalos fixos configuráveis. Sua prioridade pode também ser configurada. OBs de prioridade mais alta serão interromperão aqueles de prioridade mais baixa que estejam sendo executados.

Como os OBs chamados periodicamente são configurados



O número de OBs periódicos disponíveis depende do tipo de controlador. Um número de prioridade mais baixa representa uma prioridade de interrupção mais alta (a seleção de prioridade só está disponível nos controladores S7 400). Execução (ms) é o período de execução do OB. A defasagem de fase permite colocar os disparos em fases de interrupções periódicas entre si. A seleção de partição de imagem do processo permite que a tabela de imagem de E/S seja particionada e essa partição seja atualizada apenas quando a interrupção ocorrer (esse recurso está disponível nos controladores S7 400 apenas). O padrão é a tabela completa. No Logix, consulte a seleção de atualização de E/S de tarefa e comandos IOT.

O conteúdo de uma interrupção periódica OB geralmente lembra o conteúdo de OB1. Ele consistirá de chamadas a funções e blocos de funções que serão executados na periodicidade do OB.

Esses OBs correspondem a tarefas periódicas no Logix. Na terminologia S7 OB30 – OB38 são chamados de interrupções cíclicas OBs.

OBs de interrupção de hardware OB40 – OB47

Esses OBs podem ser configurados para disparar em um evento de entrada. Sua prioridade pode também ser configurada.

Essas são tarefas de eventos no Logix. Na terminologia S7 OB40 – OB47 são chamados de interrupções de hardware.

Por exemplo, o evento de hardware mais simples que pode ser manipulado por uma interrupção de hardware OB (ou tarefa de evento) é uma alteração de estado de uma entrada digital. Um interrupção de hardware (ou tarefa de evento) deve garantir uma resposta muito rápida à mudança.

As tarefas de evento são mais flexíveis que as interrupções de hardware OB, com disparadores não apenas de E/S, mas também de eventos de rede, instruções de programa e eventos de posicionamento.

Estrutura de programa no STEP 7

Um programa típico inclui blocos de organização (OB), blocos de funções (FB), funções (FC) e blocos de dados (DB). Os blocos de funções de sistema (SFB) e funções de sistema (SFC) geralmente estarão presentes.

- Partindo dos blocos de organização (ciclo de programa ou interrupção cíclica ou ambos), as chamadas são feitas para blocos de funções e funções.
- Um bloco de função contém código e está associado ao bloco de dados que contém os dados estáticos que o FB requer. Além dos dados estáticos, o FB tem dados temporários. FBs são usados quando a lógica deve preservar valores entre as execuções.
- Uma função contém código, mas não dados estáticos. Ela tem dados temporários. Os FCs são usados quando a lógica é concluída em uma única execução – ela não tem que preservar valores.
- Os blocos de dados são áreas para armazenagem de dados estáticos. Eles serão descritos na próxima seção.
- SFBs e SFCs são blocos de função do sistema e funções do sistema. Eles podem ser copiados de bibliotecas incluídas em uma instalação STEP 7 e colocados em um projeto.
- Quando isso tiver sido feito, eles poderão ser chamados de qualquer lugar no programa.

No STEP 7 não há estrutura equivalente à rotina/programa do Logix. O OB será a raiz da cadeia de chamada para FBs e FCs, mas como isso é feito é problema do programador.

Tarefas no Logix

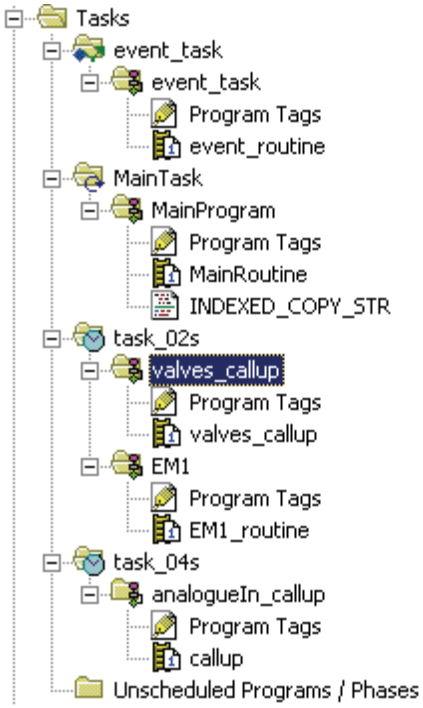
As tarefas são chamadas pelo sistema operacional. Uma tarefa oferece programação e prioridade para um ou mais programas. Cada programa contém uma seção de dados e uma ou mais rotinas de código.

As tarefas podem ser periódicas, de evento ou contínuas. A cada tarefa pode ser atribuída uma prioridade. A tarefa contínua, se presente, será sempre de prioridade mais baixa.

Um projeto Logix terá uma tarefa cujo nome padrão será MainTask. Essa tarefa pode ser contínua, periódica ou de evento. Você pode alterar seu nome se desejar.

Tarefa e estrutura do programa no Logix

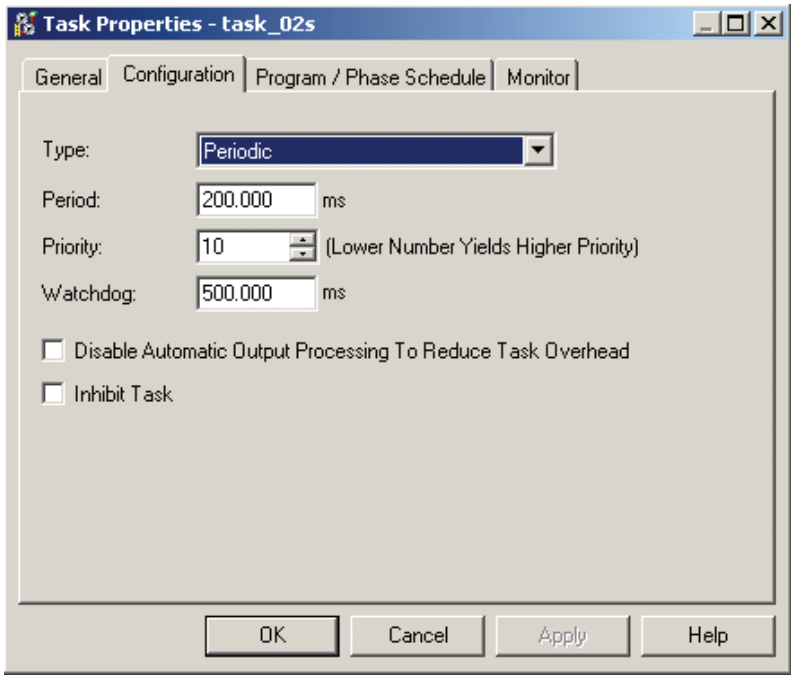
Este instantâneo de uma amostra de árvore de projeto RSLogix 5000 ajuda a ilustrar como as tarefas e programas são estruturados.



Na captura de tela acima, o ícone à esquerda de “event_task” significa uma tarefa de evento. O ícone à esquerda de “MainTask” significa uma tarefa contínua e o ícone à esquerda de “task_02s” significa uma tarefa periódica.

Tarefas periódicas

As tarefas periódicas serão disparadas em um intervalo configurado constante. A configuração do período e prioridade são mostradas abaixo.



A configuração é semelhante à página de configuração OB30 – OB38 descrita na seção “Interrupções cíclicas OB30 – OB38”.

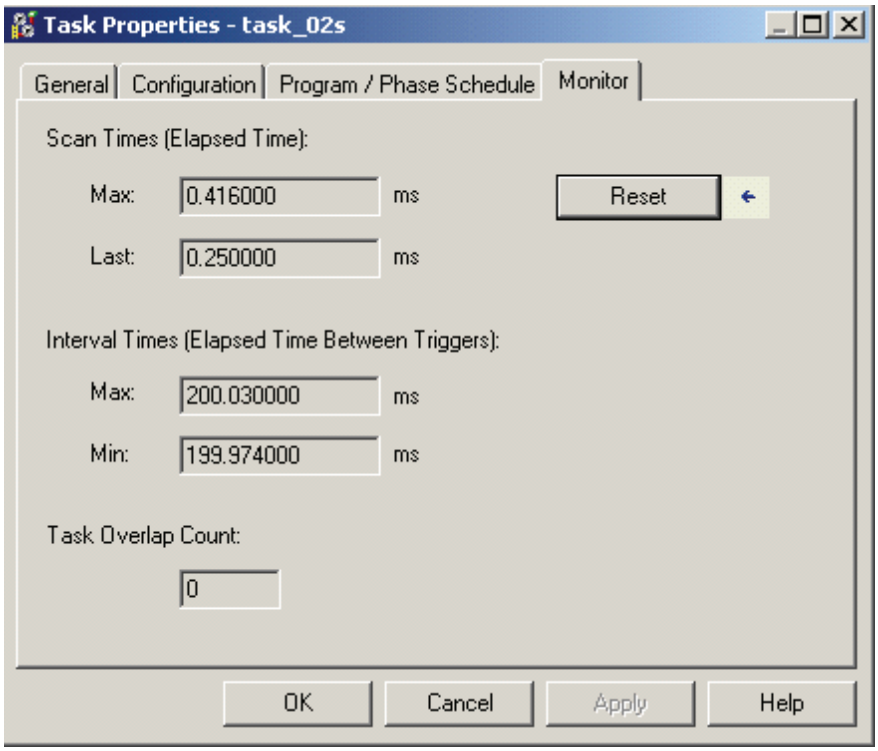
Programação de tarefas periódicas

O propósito do sistema Tarefa é:

- permitir que o programador escolha freqüências apropriadas para a execução de programas. Executando código não mais freqüentemente que o necessário, a energia das CPUs do controlador são usadas mais eficientemente para prioridades de aplicação.

- para usar o sistema de prioridade para permitir que tarefas importantes interrompam as de prioridade mais baixa, portanto aumentando a chance de execução na freqüência pretendida.

É fácil verificar esses tempo em Monitor/propriedades da tarefa.



O que acontecerá se um disparo ocorrer enquanto uma tarefa estiver sendo executada?

- Se o novo disparo for para uma tarefa com uma prioridade mais alta que aquela sendo executada, a tarefa sendo executada será interrompido pela nova, sendo retomada quando a tarefa de prioridade mais alta estiver concluída.
- Se o novo disparo for para uma tarefa com uma prioridade mais baixa que aquela sendo executada, a tarefa sendo executada continuará e a nova tarefa aguardará até que nenhuma tarefa com prioridade mais alta esteja sendo executada.
- Se o novo disparo for para uma tarefa com a mesma prioridade que aquela sendo executada, o controlador executará ambas as tarefas alternando entre elas em intervalos de 1 ms.
- Se o novo disparo for para a mesma tarefa que estiver sendo executada, o novo disparo será rejeitado. Essa é uma condição de **sobreposição**.

O número de sobreposições ocorridas desde que o contador foi reinicializado pela última vez é mostrado na janela de propriedades da tarefa. Um número diferente de zero indica que o período de interrupção precisa ser aumentado.

DICA Evite alternância desnecessária de tarefas, devido à quantidade de poder de processamento desperdiçada desnecessariamente durante a alternância.

Ao programar interrupções periódicas no Logix, observe estas semelhanças e diferenças no STEP 7:

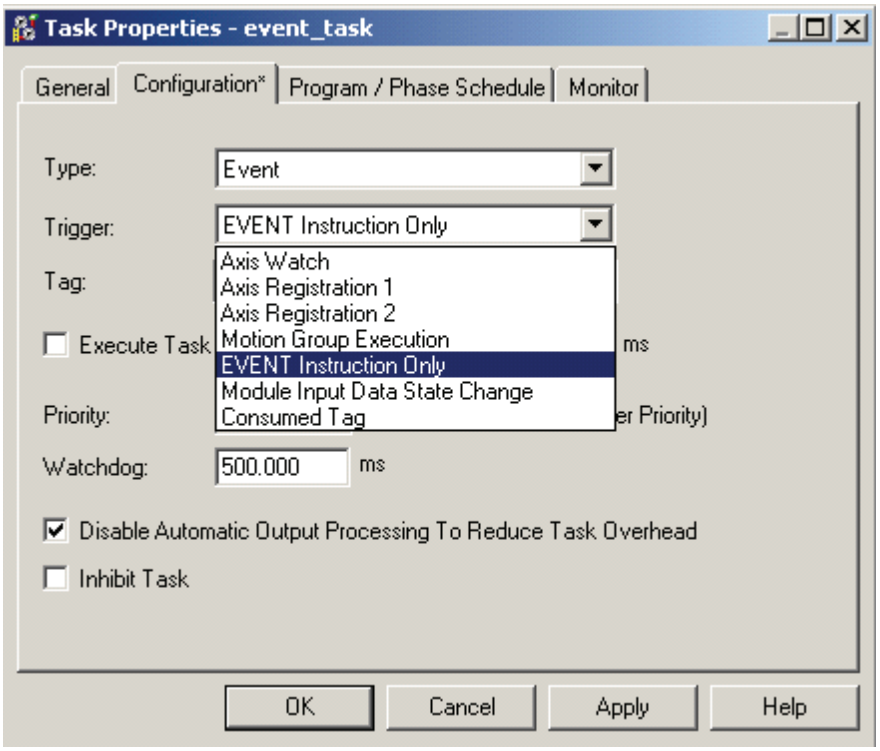
- No STEP 7, as chamadas serão feitas do OB configurado para ser executado na freqüência escolhida para as funções e blocos de funções que você deseja executar nesta freqüência. No Logix, você inserirá programas e rotinas na árvore de projeto sob a tarefa.
- No STEP 7 e Logix, o código de aplicação atual não diferirá muito do código em uma tarefa de execução contínua. Observe que a freqüência constante e conhecida de uma tarefa periódica dá aos programadores a oportunidade de transformar um simples incremento em um temporizador.
- Em ambos os sistemas, você precisará verificar as sobreposições à medida que desenvolve e testa seu código. O tempo de execução do OB ou tarefa deve ser menor do que seu período de execução.
- Verificar o tempo de execução das tarefas do Logix é fácil. Use a tela de propriedades da tarefa mostrada abaixo. No STEP 7 você precisará recolher uma amostra do relógio do sistema no início e fim do OB, subtrair os valores e armazenar o resultado em uma variável para monitoramento.

- Em um controlador S7, as sobreposições farão com que o controlador seja interrompido a menos que um OB com falha seja adicionado que detecte a falha. O Logix é menos estrito e apenas conta o número de sobreposições.

- No STEP 7, é possível organizar em fases a execução periódica dos OBs entre si. Isso não está disponível nas tarefas do Logix.

Tarefas de evento

As tarefas de evento serão executadas quando um evento de disparo configurado ocorrer. Normalmente elas terão prioridade mais alta que as tarefas periódicas.



Uma tarefa de evento é configurada abrindo-se a página Propriedades da tarefa e selecionando-se Tipo de evento. Os diferentes tipos de disparos de tarefa de evento podem ser usados para diferentes controladores Logix.

Tarefa contínua

Um controlador Logix suporta uma tarefa contínua, mas um projeto não precisa incluir a tarefa contínua. Você pode, se quiser, executar todo seu programa em tarefas de evento e periódicas.

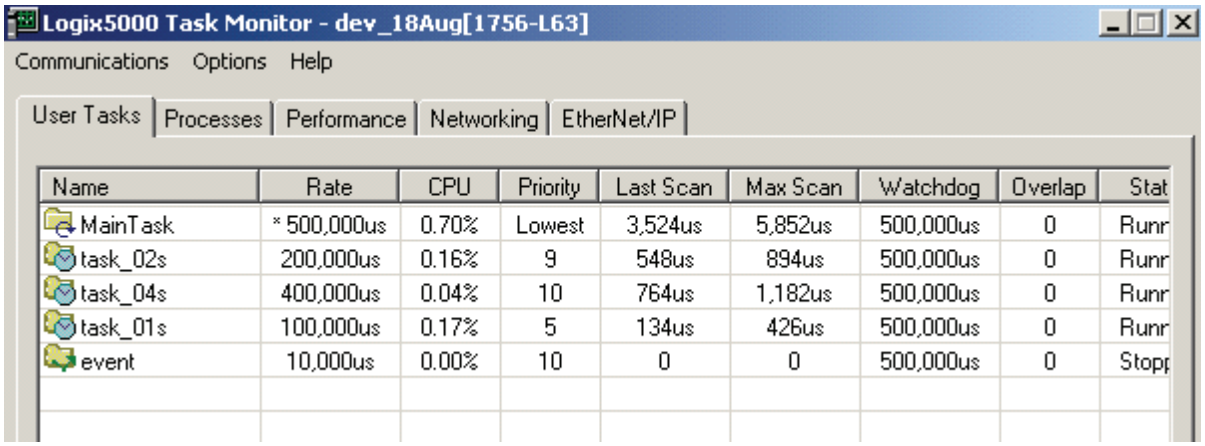
Você pode configurar se a tarefa contínua deve atualizar as saídas no final de sua execução.

Você pode, se quiser, ajustar a porcentagem do tempo de sua CPU gasto em comunicação não programável como uma porcentagem de tempo dedicada à tarefa contínua.

Monitor de tarefas

O software RSLogix 5000 inclui uma ferramenta chamada Monitor de tarefa que pode ajudar a analisar as tarefas programadas e muito mais.

A captura de tela abaixo mostra como as informações sobre as tarefas de seu controlador podem ser visualizadas em uma tabela.



As outras guias fornecem muitas informações sobre o sistema a respeito do desempenho de seu controlador. A ferramenta está incluída como padrão no disco de instalação do RSLogix 5000.

Tags não endereços

Uma das maiores diferenças que um usuário S7 notará ao começar a trabalhar com o Logix é que os dados não têm endereços. Os itens de dados são criados em um banco de dados de tags e o software RSLogix 5000 aloca endereços em segundo plano. Isso torna desnecessário que os usuários entendam e gerenciem endereços de memória. Esta seção descreve a alocação de dados nos dois sistemas.

Áreas de dados no S7

Áreas de dados nos controladores S7

Área de endereço	Notação S7	Tamanho da unidade
Tabela de entrada de imagem do processo	I	Bit de entrada
	IB	Byte de entrada
	IW	Palavra de entrada
	ID	Palavra dupla de entrada
Tabela de saída de imagem do processo	Q	Bit de saída
	QB	Byte de saída
	QW	Palavra de saída
	QD	Palavra dupla de saída
Memória de bit	M	Bit de memória
	MB	Byte de memória
	MW	Palavra de memória
	MD	Palavra dupla de memória
Temporizadores	T	
Contadores	C	
Bloco de dados	DBX	Bit de dados
	DBB	Byte de dados
	DBW	Palavra de dados

As seções abaixo dizem mais sobre as duas áreas mais comumente usadas na programação — memória de bits e blocos de dados.

Memória de bit

As locações de "memória de bits" são denotadas por M*x*, por exemplo:

- **M5.3** é um bit.
- **MB6** é um byte (BYTE).
- **MW8** é uma palavra de 16 bits (WORD).
- **MD10** é uma palavra de 32 bits (DWORD).

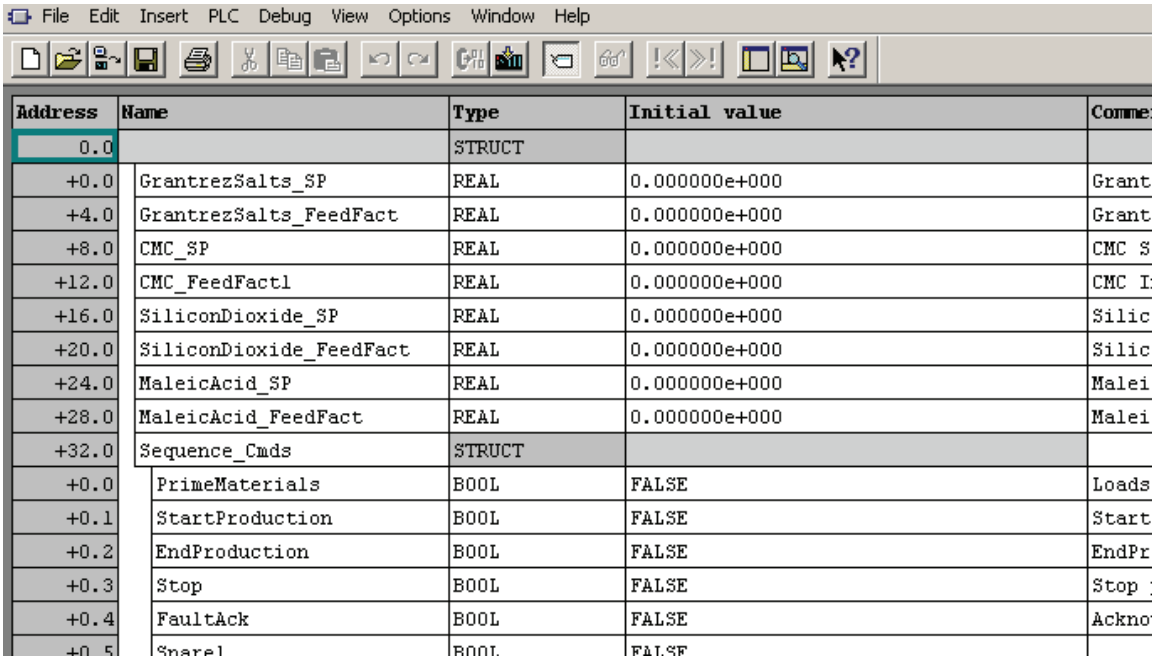
As locações de memória de bits podem ser rotuladas na tabela de símbolos (semelhante à tabela de símbolos CLP-5 ou SLC), conforme mostrado na captura de tela a seguir.

Status	Symbol	Address	Data type	Comment
	EXT_ZONE2_ON	Q 28.2	BOOL	EXTRUDER ZONE2 ON
	EXT_ZONE3_ON	Q 28.3	BOOL	EXTRUDER ZONE3 ON
	EXT_ZONE4_ON	Q 28.4	BOOL	EXTRUDER ZONE4 ON
	EXT_ZONE5_ON	Q 28.5	BOOL	EXTRUDER ZONE5 ON
	FALSE	M 0.0	BOOL	
	FlowTotaliser	UDT 10	UDT 10	
	FSL24001	I 10.0	BOOL	
	FSL24002	I 41.0	BOOL	
	FSL24003	I 10.6	BOOL	
	FT24001	PIW 530	INT	
	FT24002	PIW 526	INT	
	FT24006	PIW 570	INT	
	GET_INDEXED_REFE...	FC 111	FC 111	
	Global_Data	DB 99	DB 99	
	IFIX_alarms	DB 71	DB 71	
	INDEXED_COMPARE	FC 102	FC 102	
	INDEXED_COPY	FC 101	FC 101	
	Interlocks_Handler	FB 70	FB 70	
	Interrupt_Execution	OB 35	OB 35	
	JUNK_BIT	M 0.2	BOOL	
	KTRON_CmdsFromM...	DB 33	DB 33	Commands & Setpoints From Manufacturing PLC
	KTRON_StatusToMan...	DB 32	DB 32	Feeder Status to ManufacturingPLC
	LF24001A	Q 17.0	BOOL	PIG LAUNCH VALVE
	LF24002A	Q 18.5	BOOL	PIG LAUNCHER
	LF24003A	Q 10.4	BOOL	PIG LAINCHER

Blocos de dados

Os blocos de dados têm status semelhante a outros blocos – blocos de organização, blocos de funções e funções – exceto por conterem dados em vez de código de programa. A memória nos blocos de dados é estática – os dados mantêm seu valor até serem alterados.

Exemplo de um bloco de dados



Address	Name	Type	Initial value	Comme
0.0		STRUCT		
+0.0	GrantrezSalts_SP	REAL	0.000000e+000	Grant
+4.0	GrantrezSalts_FeedFact	REAL	0.000000e+000	Grant
+8.0	CMC_SP	REAL	0.000000e+000	CMC S
+12.0	CMC_FeedFactl	REAL	0.000000e+000	CMC I
+16.0	SiliconDioxide_SP	REAL	0.000000e+000	Silic
+20.0	SiliconDioxide_FeedFact	REAL	0.000000e+000	Silic
+24.0	MaleicAcid_SP	REAL	0.000000e+000	Malei
+28.0	MaleicAcid_FeedFact	REAL	0.000000e+000	Malei
+32.0	Sequence_Cmds	STRUCT		
+0.0	PrimeMaterials	BOOL	FALSE	Loads
+0.1	StartProduction	BOOL	FALSE	Start
+0.2	EndProduction	BOOL	FALSE	EndPr
+0.3	Stop	BOOL	FALSE	Stop
+0.4	FaultAck	BOOL	FALSE	Ackno
+0.5	Spare1	BOOL	FALSE	

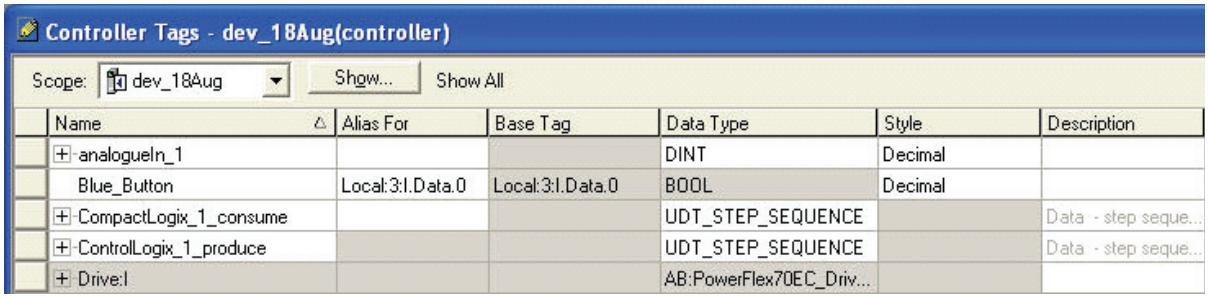
Os símbolos do bloco de dados não aparecem na tabela de símbolos, mas o nome do bloco de dados sim.

Os blocos de dados podem ser atribuídos para manter os dados usados pelos blocos de funções. Esses são chamados blocos de dados de instância.

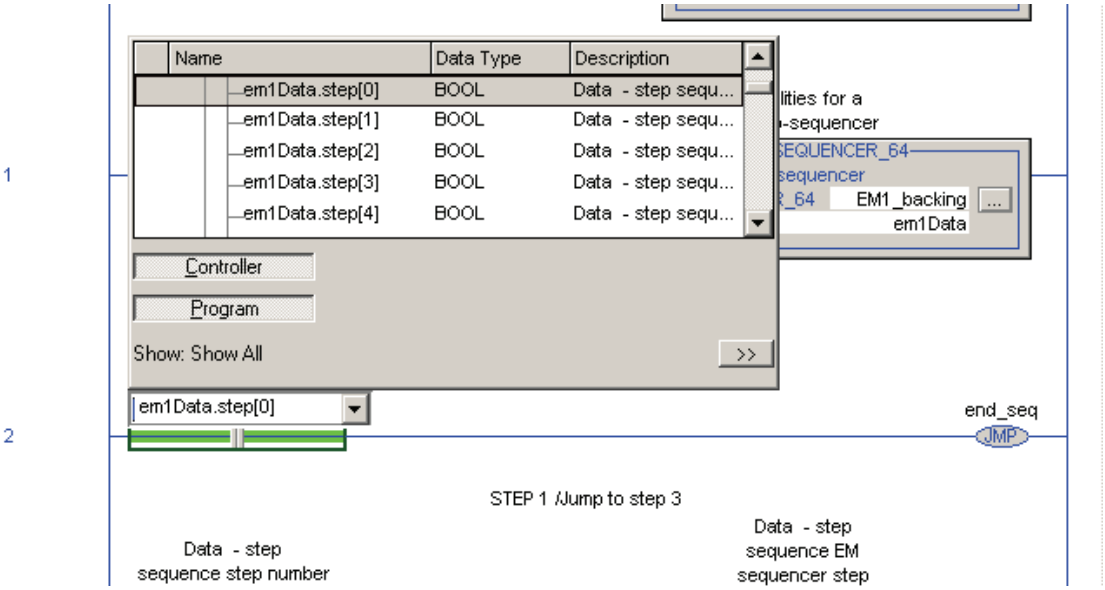
Dados no Logix

No ambiente de programação RSLogix 5000, os dados são configurados em um banco de dados de tags. Os endereços de memória ficam ocultos, o que torna as coisas mais fáceis para o programador.

Banco de dados de tags



Selecionar um tag do menu suspenso ao programar



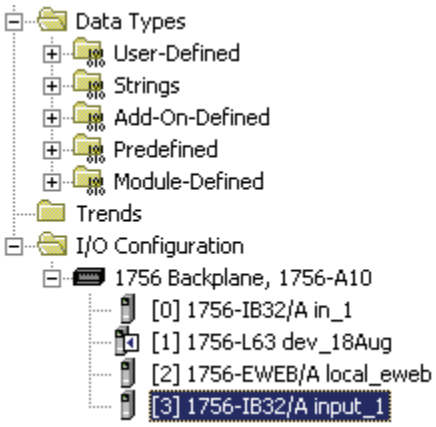
No Logix, há um **banco de dados de tags do escopo do controlador** e **bancos de dados de tags de escopo do programa** associados a cada programa.

- Os tags no banco de dados do escopo do controlador são globais e podem ser acessados por rotinas em qualquer parte do programa.
- Os tags do escopo do programa só podem ser acessados por rotinas nesse programa.

E/S e tags de alias

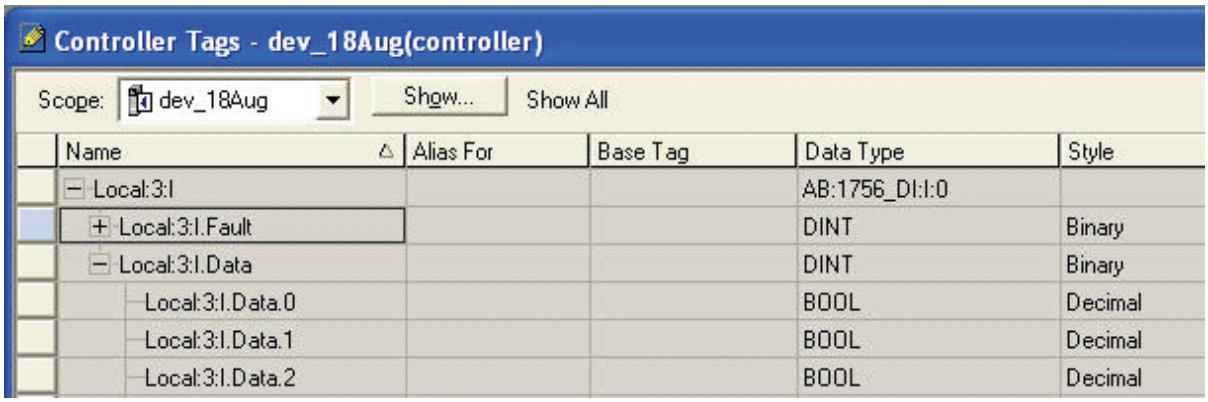
Um tag de alias permite que você represente outro tag, pois ambos os tags têm o mesmo valor. Uma das finalidades dos aliases é fazer referência aos tags de E/S conforme descrito abaixo.

Os módulos de E/S podem ser adicionados a um projeto pela adição do módulo ao backplane do controlador na pasta do projeto.



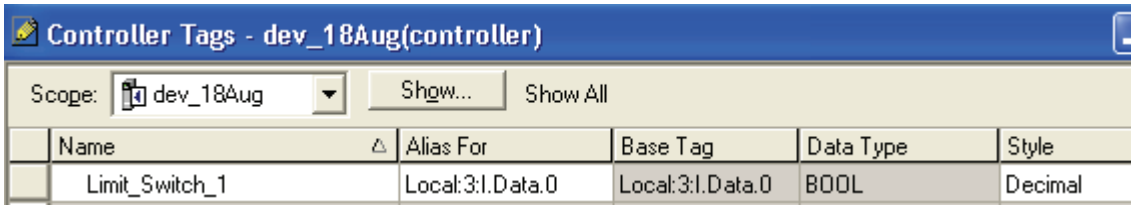
Nesse caso, um cartão de entrada de 32 pontos foi adicionado ao slot 3. O número do slot está entre colchetes no início da linha. "1756-IB32/A" é o número de peça do cartão. "entrada_1" é um nome para o cartão configurado quando o cartão é adicionado ao rack pela primeira vez.

Após adicionar o cartão, o software RSLogix 5000 gerará automaticamente os tags de perfil do dispositivo relevante ao banco de dados de tags do escopo do controlador. Eles são os tags de entrada Local:3:I e de configuração Local:3:C abaixo.



Name	Alias For	Base Tag	Data Type	Style
Local:3:I			AB:1756_DI:I:0	
Local:3:I.Fault			DINT	Binary
Local:3:I.Data			DINT	Binary
Local:3:I.Data.0			BOOL	Decimal
Local:3:I.Data.1			BOOL	Decimal
Local:3:I.Data.2			BOOL	Decimal

Você pode criar um novo tag de alias com um nome mais descritivo. Por exemplo, um alias para a primeira entrada pode ser criado, chamado Limit_Switch_1, que descreve fisicamente essa entrada.



No STEP 7, a ferramenta de configuração de hardware atribuirá endereços a um cartão de E/S quando este for adicionado ao sistema. Por exemplo, um cartão de entrada digital pode ter os bytes I16 e I17 atribuídos. Então o programador identificará os endereços de bits de cada entrada e inserirá um nome para ela na tabela de símbolos. Após terminar, o programa fará automaticamente a associação I16.5 = “ZSC2036”.

Linguagens de programação

Esta seção descreve as linguagens de programação disponíveis com o software STEP 7 e RSLogix 5000. Todas as linguagens não são padrão; elas dependem da versão do software comprado. A seleção do idioma Logix mais adequado à tarefa resultará em um projeto de programa mais fácil, codificação mais rápida e um programa mais fácil de compreender.

Há uma diferença significativa entre as linguagens S7 e Logix. No S7, a lista de declarações é a linguagem "nativa" do controlador. Outras linguagens são convertidas em STL. No Logix, todas as linguagens são "nativas" no controlador – cada uma é compilada sem referência a qualquer uma das outras. O benefício disso é que quando você carrega um programa do controlador, você o visualiza na linguagem em que ele foi escrito.

O **STEP 7** tem três linguagens padrão:

- Lista de declarações (STL) – pode ser descrito como montador de alto nível.
- Lógica ladder (LAD)
- Diagrama de bloco de funções (FBD)

E algumas linguagens opcionais:

- Texto estruturado (ST)
- CFC – Fluxograma contínuo para aplicações de tipo de processo
- HiGraph – Controle seqüencial via software Graphing
- ML – Linguagem de posicionamento – semelhante ao GML no antigo controlador dedicado Rockwell Automation 1394

Um programa pode consistir em blocos de funções e funções escritas em diferentes linguagens.

O software RSLogix 5000 tem quatro linguagens de programação:

- Diagrama de lógica ladder (LD) – comparável a LD Siemens, com um conjunto de instruções expandido.
- Texto estruturado (ST) – equivalente ao ST Siemens.
- Diagrama de bloco de funções (FBD) – equivalente ao CFC Siemens.
- Controle seqüencial de funções (SFC) – comparável ao hiGraph Siemens.

Uma rotina – a seção básica de código em Logix – pode estar em qualquer uma dessas linguagens e um programa pode ser feito de rotinas escritas em diferentes linguagens. A captura de tela a seguir mostra um exemplo.

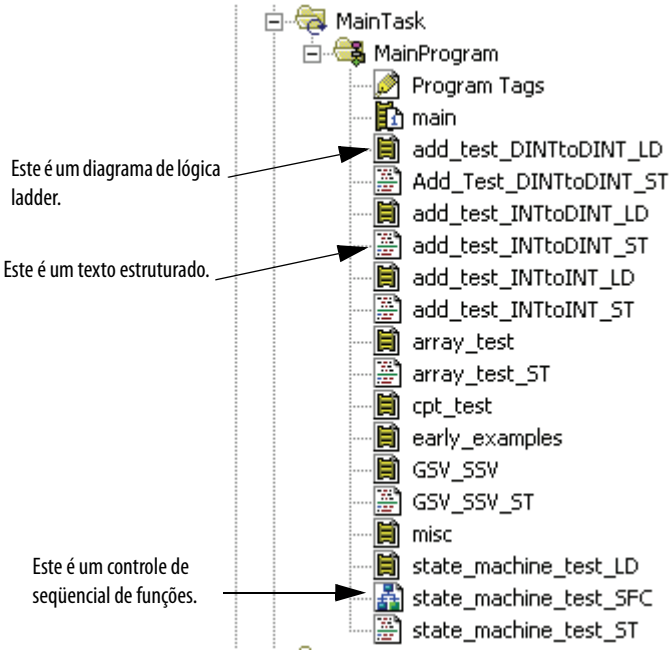


Diagrama de lógica ladder Logix

Tradicionalmente, o diagrama de lógica ladder é usado para implementar lógica combinacional booleana. No Logix, isso também pode ser usado para lógica seqüencial, posicionamento, manipulação de dados e cálculos matemáticos, apesar de que outras linguagens podem ser mais convenientes para essas tarefas.

Texto estruturado do Logix

Texto estruturado é uma linguagem de procedimento de alto nível fácil de aprender por qualquer um com experiência em Basic, Pascal ou uma das linguagens da família 'C'. É usado principalmente para manipulação de dados e cálculos matemáticos, embora a lógica seqüencial, combinacional e de posicionamento possa ser facilmente programada em ST.

Diagrama de blocos de funções Logix

O diagrama de blocos de funções descreve graficamente uma função (booleana ou matemática) relacionada a variáveis de entrada ou variáveis de saída. As variáveis de entrada e saída estão conectadas a blocos por linhas de conexão. Uma saída de um bloco pode também estar conectada a uma entrada de outro bloco.

É uma boa prática programar loops PID no FBD. Essa é a linguagem mais conveniente para controle de processo.

Controle seqüencial de funções Logix

SFC é uma ferramenta gráfica para descrição de lógica seqüencial como um conjunto de estados e transições. As saídas podem ser atribuídas a um estado e as condições boolenas para transições a outros estados definidos.

Conversão de código STEP 7 em Logix

- Se você tiver código de lógica ladder STEP 7 que queira converter em Logix, LD deve ser sua primeira escolha. O significado do LD é semelhante em ambos os sistemas.
- Se você tiver código de diagrama de blocos de funções STEP 7 que queira converter em Logix, FBD deve ser sua primeira escolha.
- Observe que o FBD Logix padrão é mais avançado que o FBD STEP 7 e é equivalente à linguagem CFC STEP 7 opcional.
- Se você tiver um código de lista de declarações STEP 7 que queira converter em Logix, a linguagem mais adequada dependerá da natureza do bloco STL. Se o bloco STL contiver principalmente avaliações boolenas, LD será provavelmente a melhor linguagem Logix para conversão. Se o bloco STL contiver indicadores para acessar e manipular dados ou executar cálculos matemáticos, ST será provavelmente a melhor linguagem Logix para conversão. Se o bloco STL contiver lógica seqüencial, SFC deve ser considerada, embora a lógica seqüencial possa ser facilmente implementada em ST e LD.

Vetores não indicadores

No STEP 7, os vetores podem ser definidos exatamente como seriam em Pascal ou C, mas as linguagens básicas (STL, LD e FBD) não oferecem suporte de alto nível para acessá-los. Em vez disso, rotinas de indicadores devem ser criadas.

As funções de biblioteca STEP 7 não têm suporte para acesso de vetor.
Os programadores que dominarem os indicadores podem escrever suas próprias funções, como FC101 “INDEXED_COPY” (veja abaixo), mas isso requer habilidade e tempo.

“INDEXED_COPY” no STEP 7 faz o mesmo que a instruções COP do Logix para cópia indexada.

```
CALL  "INDEXED_COPY"                FC101
 indexSrc:=#index_in
 source  :="Instance_FB2".table     P#DB4.DBX0.0
 indexDst:=1
 dest    :="Instance_FB2".target    P#DB4.DBX96.0
 len     :=8
```

O FC111 abaixo acessará um vetor.

```
CALL  "GET_INDEXED_REFERENCE"       FC111
 refArray  :="Instance_FB2".table   P#DB4.DBX0.0
 index     :=#index_in
 byteIncr  :=32
 startIndex:=TRUE
 retVal    :=#ptr
```

O indicador para o objeto é retornado no parâmetro #ptr, que pode ser, então, desreferenciado para obter os dados.

No Logix, os vetores podem ser ambos definidos e acessados da maneira comum de uma linguagem de computador de alto nível, como mostra o fragmento abaixo.

```
// copy a string from a table of strings #table
// to a target string #target. The index is #index_in

COP(table[index_in], target, target.LEN);
```

Instruções add-on

Resumo de instruções add-on

As instruções add-on são o equivalente dos blocos de funções STEP 7, com dados privativos e escolhas de parâmetro avançadas.Em particular, o tipo de parâmetro INOUT ou "passado por referência" torna possível passar estruturas de dados eficientemente ao código.

Como a instrução add-on é muito semelhante ao bloco de funções do STEP 7, é provável que o programador S7 que esteja convertendo em Logix fará uso dela prontamente.

Comparação entre os FBs e instruções add-on:

- Ambas podem ser chamadas como funções nomeadas a partir de qualquer lugar no programa.
- Ambas contêm uma área de dados privativos de dados estáticos, embora ela não seja verdadeiramente privativa no caso do STEP 7.
- Um bloco de funções STEP 7 também tem uma área de dados temporários.
- Na instrução add-on, os dados estáticos locais farão a mesma coisa.

Ambos têm três tipos de parâmetros – input (passado por valor), output (passado por valor) e in-out (passado por referência). O parâmetro passado por referência é um benefício considerável, uma vez que permite que grandes estruturas de dados sejam passadas eficientemente.

A instrução add-on manterá automaticamente um histórico de alterações gravando o registro de data e hora e o nome de usuário do Windows no momento da alteração. Isso não está disponível nos blocos de funções do STEP 7.

Com a instrução add-on uma rotina de pré-varredura pode ser configurada para ser executada quando o controlador passar do modo Programa para o modo Executar ou for ligado no modo Executar. Sob essas condições, a rotina de pré-varredura será executada uma vez e geralmente pode ser usada para inicializar dados. No STEP 7 o bloco de organização OB100 faz o mesmo, mas o código de pré-varredura não pode ser anexado especificamente a um FB.

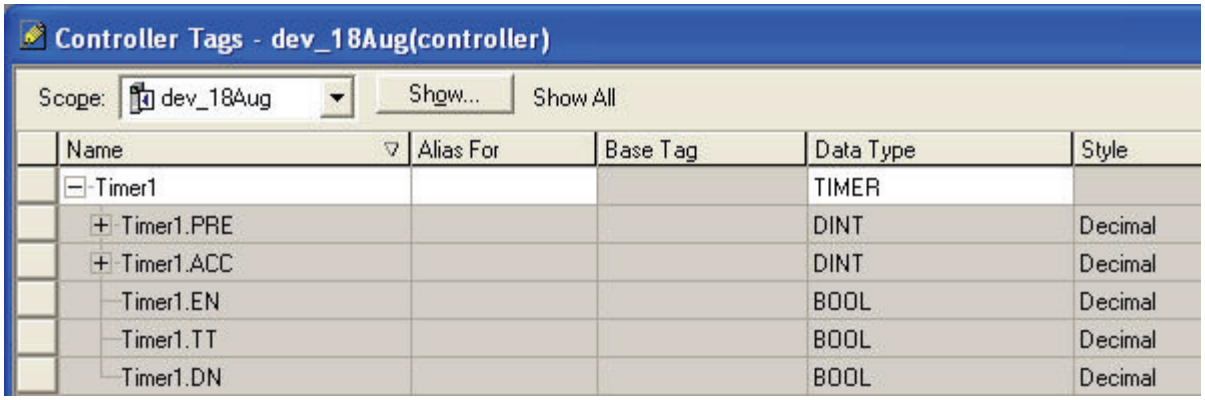
Se a instrução add-on for chamada de uma etapa SFC e o SFC for configurado para reinicialização automática, uma rotina de pós-varredura definida na instrução add-on será executada uma vez quando o SFC sair dessa etapa. Isso pode ser usado para reinicializar dados. Um FB STEP 7 não tem equivalente incorporado (embora seja fácil programar um).

Uma instrução add-on pode ter uma rotina EnableInFalse, que será chamada (se presente) quando a condição de linha na chamada da instrução add-on for falsa. Nesse caso, os parâmetros de entrada e saída passarão valores. Um FB STEP 7 não tem equivalente.

As instruções add-on são exploradas com mais detalhes no Capítulo 4.

Tags de backing

Muitas instruções e tipos de dados usam tags de backing – tags criados especificamente para a instância da instrução ou tipos de dados que você está instanciando. Instruções add-on, temporizadores, contadores, mensagens e controle PID usam tags backing. O software RSLogix 5000 gerará a estrutura correspondente de elementos em qualquer momento em que você crie tag desse tipo de modo que você não tenha que criar os elementos.



O Common Industrial Protocol (CIP)

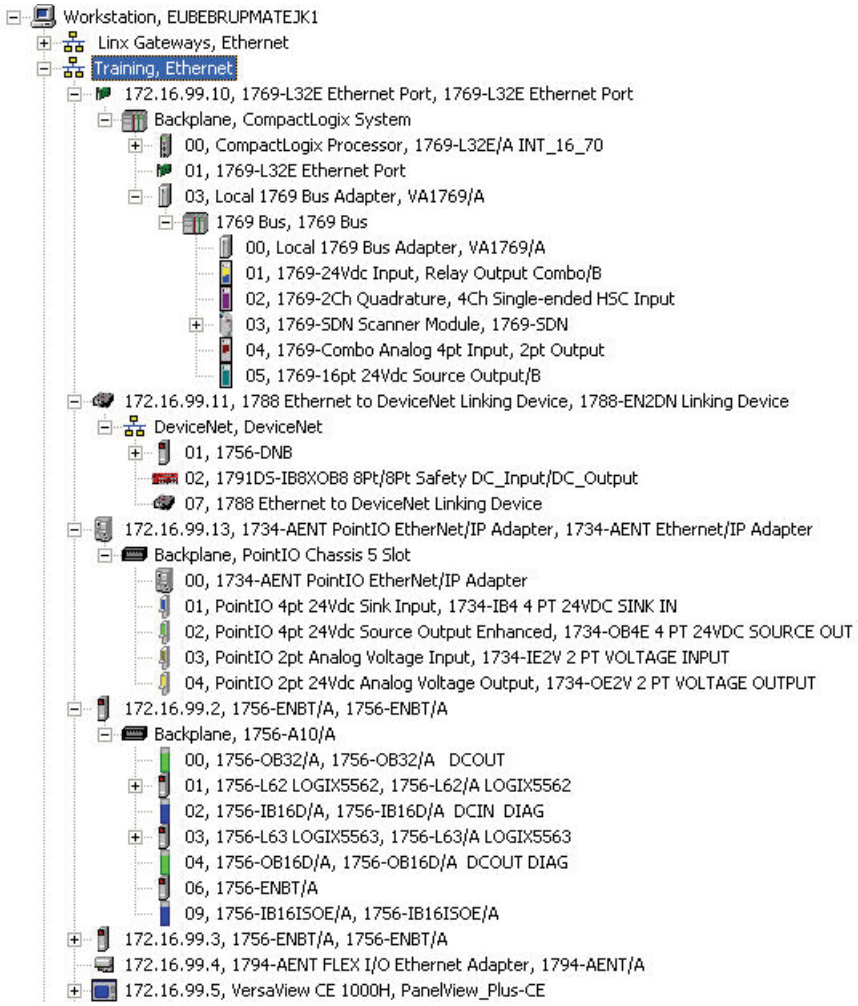
O Logix usa três redes principais — Ethernet/IP, ControlNet e DeviceNet. Cada uma tem características adequadas a diferentes áreas da aplicação. Os três tipos de redes compartilham um protocolo, o 'Common Industrial Protocol'.

O CIP torna possível transferir dados por meio de qualquer um dos três tipos de redes suportados pelo Logix com uma configuração e interface de programação praticamente idêntica para os três. Além disso, os dados podem ser transferidos por meio de uma rede construída a partir de mais de um dos três tipos de rede sem qualquer necessidade do programador converter protocolos.

No S7 "tradicional" os dois protocolos principais são Ethernet industrial, para rede de TI e para outros controladores, e Profibus DP para rede para sistemas de campo. Esses dois protocolos são separados no nível do hardware e no nível de dados. Com o hardware e software S7 mais recentes, "Profinet CBA" integra Ethernet industrial, Profinet e Profibus.

Visualizando a rede

Os usuários S7 podem achar a configuração e gerenciamento da rede Logix impressionantes. Como exemplo, a árvore abaixo mostra os dispositivos realmente conectados ao sistema. Essa árvore foi criada entrando em comunicação – nada foi configurado.



A redes são descritas mais adiante no Capítulo 1.

Intercâmbio de dados entre controladores

Envio/recebimento no STEP 7

Para preparar a comunicação controlador para controlador no STEP 7, estas etapas são seguidas.

1. As estações remotas são configuradas graficamente em um componente STEP 7 chamado NetPro.
2. Uma tabela de conexão é criada no NetPro especificando os protocolos e parâmetros de cada conexão.
3. As funções de biblioteca FC5 AG_SEND e FC6 AG_RECV são copiadas no projeto.
4. As chamadas são feitas do programa do usuário para AG_SEND e AG_RECV, especificando parâmetros de configuração e as áreas de dados usadas para gerar e receber dados.

Tags produzidos/consumidos no Logix

Tags produzidos e consumidos são o modo como dados importantes são transferidos entre os controladores Logix em rede em cada período definido. Os tags produzidos e consumidos podem transmitir por Ethernet/IP ou ControlNet e no backplane dos controladores ControlLogix.

Os tags produzidos e consumidos são aqueles configurados como produzidos ou consumidos durante a criação. Se um tag for marcado como produzido, então seu valor será transmitido para uma rede EtherNet/IP ou ControlNet a qual o controlador esteja conectado. Se marcado como consumido, então o controlador do qual o tag requer dados será identificado como parte da configuração e o tag consumido receberá seu valor do tag produzido equivalente nesse controlador.

Há canais separados para envio e recebimento. Alterar o valor de um tag de consumo não terá efeito na produção do tag. Isso remete à comunicação controlador para controlador no S7 e difere da comunicação controlador para SCADA, em que qualquer alteração se refletirá na outra extremidade.

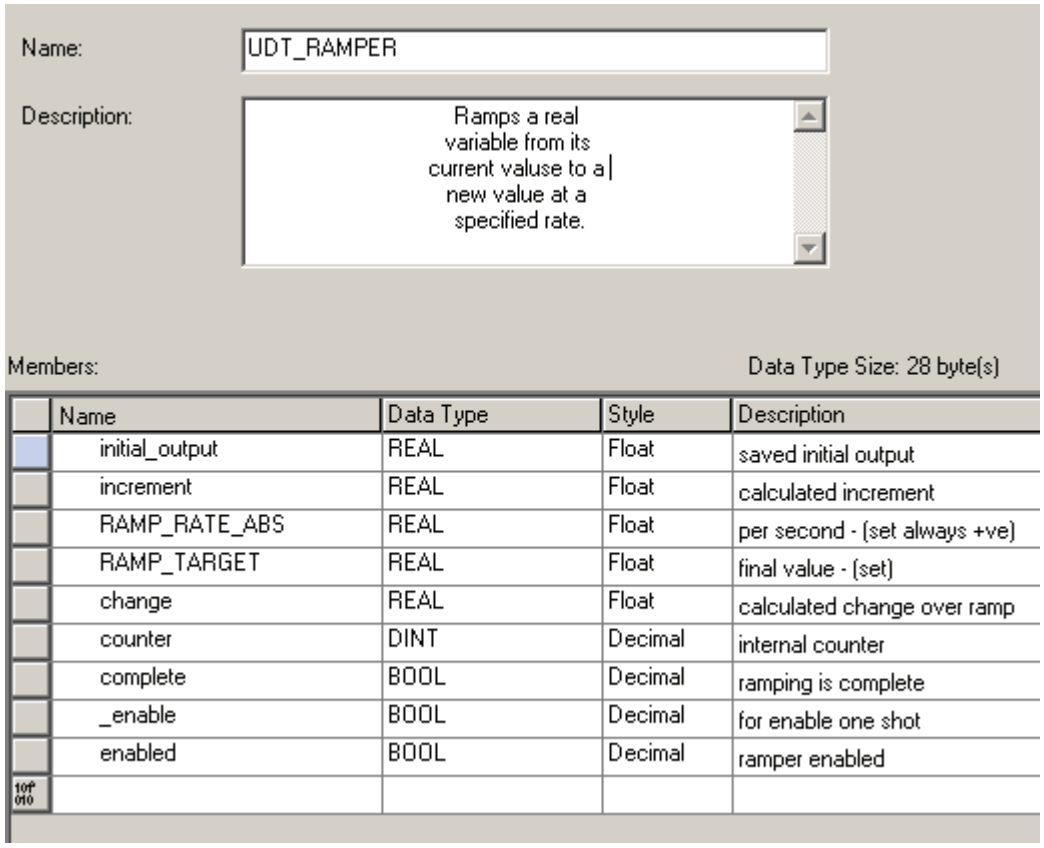
Nenhuma programação é necessária para configurar as conexões de produção/consumo. Isso contrasta com o S7, no qual alguma codificação é necessária para comunicações controlador para controlador (ENVIO/RECEBIMENTO).

Tipos de dados definidos pelo usuário

No Logix, os tipos de dados definidos pelo usuário podem ser configurados. Isso permite que a estrutura de um tipo de dados complexo seja declarada como um tipo. Instâncias desse tipo podem, então, serem definidas no programa.

Os tipos de dados definidos pelo usuário Logix têm configuração e utilização muito semelhante aos tipos de dados definidos pelo usuário STEP 7.

UDT Logix



Name	Data Type	Style	Description
initial_output	REAL	Float	saved initial output
increment	REAL	Float	calculated increment
RAMP_RATE_ABS	REAL	Float	per second - (set always +ve)
RAMP_TARGET	REAL	Float	final value - (set)
change	REAL	Float	calculated change over ramp
counter	DINT	Decimal	internal counter
complete	BOOL	Decimal	ramping is complete
_enable	BOOL	Decimal	for enable one shot
enabled	BOOL	Decimal	ramper enabled

Atualização de E/S assíncronas

Nos sistemas Logix, a E/S é atualizada de maneira assíncrona no que diz respeito aos períodos de execução do programa, em contraste com a abordagem CLP tradicional usada no S7, na qual uma tabela de imagem E/S é atualizada no início do ciclo e os valores de entrada não mudam durante uma execução do programa.

O programador Logix precisará considerar se há alguma necessidade de colocar os dados de entrada em buffer, de modo que seu valor permaneça constante durante a execução do programa.

É muito comum "consumir" entradas apenas passando-as como parâmetros a um módulo do código. As entradas não serão usadas em mais nenhum lugar no programa. Isso elimina qualquer necessidade de buffer. Veja o exemplo de módulo de controle no Capítulo 4.

O tipo de dados DINT

Os controladores Logix operam em tags DINT (inteiro de 32 bits) mais eficientemente que em INT (inteiro de 16 bits) ou SINT (inteiro de 8 bits). Use DINT sempre que possível, mesmo se a faixa de valores em que você está trabalhando se encaixe em INT ou SINT. Esses tipos de dados são fornecidos por motivos de compatibilidade com IEC61131-3, mas são convertidos internamente em DINTS antes de serem usados pelo programa, de modo que o código será executado mais eficientemente na maioria das situações.

Gerenciador de fases

Gerenciamento de fases no STEP 7

O STEP 7 não possui ferramentas incorporadas para executar o gerenciamento de fases. As estruturas necessárias devem ser programadas em um conjunto de rotinas, geralmente referenciadas como PLI ou interface lógica de fase. Os componentes de um programa PLI com base em S88 são:

- Um seqüenciador de etapas cujo comportamento está em conformidade com o modelo de estado S88. Determinadas etapas ou faixas de etapas definem o estado S88. Os comandos do seqüenciador são também especificados por S88 e o seqüenciador responderá apenas quando o modelo de estado permitir. Um seqüenciador com essas propriedades é chamado de fase.

- Um conjunto de dados de cada fase é usado para registrar o status da fase e receber comandos do gerenciador de receitas. O gerenciador de receitas se comunica com esses dados. O formato dos dados dependerá do gerenciador de receitas.

- Um módulo de lógica que converte o status da fase no formato requerido pelo gerenciador de receitas e converte o comando do gerenciador de receitas em comandos de fase.

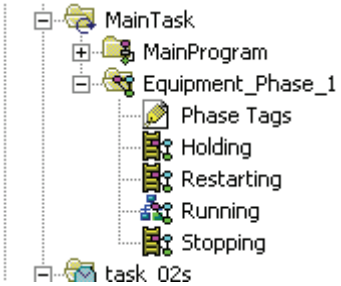
Gerenciador de fases no Logix

Em uma fase de equipamento S88, há estados especificados da fase, bem como as transições entre esses estados. O gerenciador de fases é uma funcionalidade do software RSLogix 5000 que permite fazer três coisas:

- Alocar o código de cada estado de fase para uma rotina diferente.
- Executar uma máquina de estados em segundo plano que lide que as transições entre estados da fase.
- Gerenciar a execução da fase usando um conjunto de comandos Logix.

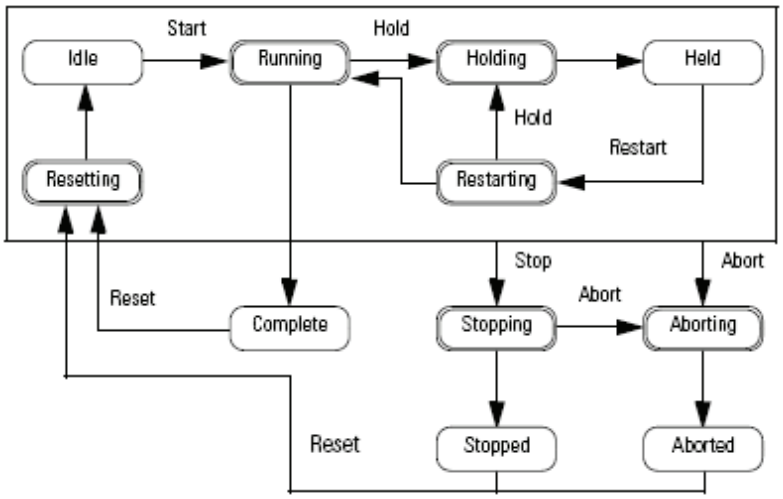
Ele é usado em uma variedade de espaços de aplicação, incluindo, mas não limitado a, controle e empacotamento de processo, porque isso permite clara separação de controle de dispositivos/equipamentos e de controle de procedimentos, portanto tornando a criação de código muito mais modularizada e eficiente, especialmente na padronização de grandes sistemas.

Fase de equipamento de fase na árvore de projeto



O código de cada estado da fase pode ser escrito em qualquer das linguagens Logix.

Essa é a máquina de estados de fase. Ela é quase idêntica ao modelo de estados S88.



Se você tiver programado um gerenciador de fases/rotina PLI S88 compatível com STEP 7 e desejar convertê-lo em Logix, pode ser possível evitar a conversão usando o gerenciador de fases do Logix.

Tempo de sistema (CST)

O S7 tem um relógio de sistema, que é representado usando-se 32 bits e contagens em milissegundos. Seu valor pode ser obtido (e armazenado) fazendo-se uma chamada ao sistema operacional, o que é útil para medição precisa dos intervalos de tempo.

O Logix usa o tempo de sistema, que é um número de 64 bits que mede o número de microssegundos desde a última inicialização do controlador. Como no S7, os intervalos podem ser medidos fazendo-se chamadas ao sistema operacional para obter o valor do tempo de sistema. Ele oferece a base da sincronização do relógio para sistemas com várias CPUs, funcionalidade de controle preciso de posicionamento, chaveamento de saída programável preciso até 100 µs, registro de data e hora de evento de entrada, amostragem analógica programável, monitoração e comunicação de E/S de segurança, cálculos de posição de came de posicionamento e tempo de wallclock.

Entradas com registro de data e hora

O registro de data e hora é uma funcionalidade que registra uma alteração nos dados de entrada com um tempo relativo de quando a alteração ocorreu. Com os módulos de entrada digital, você pode configurar um registro de data e hora das alterações de dados. Você pode usar o registro de data e hora CST para comparar o tempo relativo entre as amostras de dados.

Isso permite que o programador consiga uma precisão incomparável ao vincular sinais de entrada para aplicações como o controle de posicionamento comumente usado, sem colocar uma grande carga sobre os sistemas de comunicação e processamento de lógica e o código de aplicação relacionado.

Saídas programadas

Com os módulos de saída digital, você pode configurar o módulo para definir as saídas em um tempo programável.

Isso permite que o programador consiga uma precisão incomparável ao vincular saídas às referências de tempo de aplicações como posições de eixo no controle de posicionamento ou funções de controle de processo, sem colocar uma pesada carga sobre os sistemas de comunicação e processamento de lógica e o código da aplicação relacionado.

Sem variáveis temporárias

O S7 tem uma categoria de variáveis chamada de variáveis temporárias. Seu escopo é o bloco de programa no qual elas são definidas e seu tempo de vida é a execução do bloco de programa no qual foram definidas.

O Logix não tem um equivalente para as variáveis temporárias. Todas as variáveis são estáticas – elas mantêm seus valores até serem alteradas.

Para obter a funcionalidade geralmente conseguida nas aplicações S7, use, por exemplo, uma das seguintes abordagens:

- Use tags do escopo do programa.
- Se você estiver programando a instrução add-on, use os tags locais (parte dos dados da instrução add-on).

Sem necessidade de acumuladores ou registradores especiais

Se você programa usando a lista de declarações do STEP 7, você está familiarizado com os acumuladores e os registradores de indicador AR1 e AR2. Não há equivalentes no Logix. Todos os operadores são tags.

Para obter a funcionalidade geralmente conseguida nas aplicações S7, use, por exemplo, uma das seguintes abordagens:

- Use tags do escopo do programa.
- Se você estiver programando uma instrução add-on, use os tags locais (parte dos dados da instrução add-on).
- Considere se você precisa de equivalentes Logix dos acumuladores e registradores especiais do S7. Eles estão lá devido à natureza de baixo nível das listas de declarações do S7 e em uma linguagem como o texto estruturado, é improvável que isso seja necessário.

Conversão de software do sistema e funções padrão

Introdução

Este capítulo relaciona as funções do sistema S7 mais comumente usadas, explica como o equivalente é feito no Logix e fornece diversos exemplos específicos.

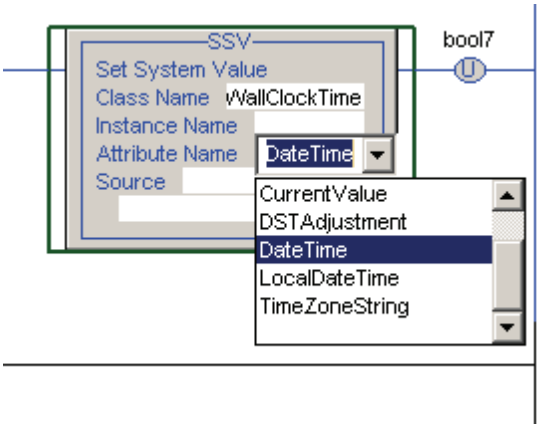
Tópico	Página
Funções do sistema Logix	68
Copiar	68
Configuração e leitura de data e hora	69
Leitura de hora do sistema	69
Tratamento de interrupções	70
Erros	70
Status – controlador	71
Status – módulo	71
Status – for OBs e tarefas	72
Temporizadores	72
Rotinas de conversão	73
Rotinas de manuseio de grupo	73
Exemplos de chamadas de função do sistema	74

A finalidade deste capítulo é torná-lo ciente das instruções dedicadas disponíveis no Logix, de modo que você não desperdice tempo desenvolvendo soluções que já existem.

Funções do sistema Logix

No Logix, o equivalente da maioria das funções do sistema S7 são as instruções GSV (Get System Value) e SSV (Set System Value). Essas instruções acessam uma hierarquia de objetos (classes, instâncias e atributos) incorporadas aos controladores do Logix. Se você programar GSV e SSV, menus suspensos o guiarão através da seleção de parâmetros.

Instrução SSV



Depois que os fundamentos básicos de GSV e SSV tiverem sido aprendidos, o novo usuário Logix pode achar que o acesso ao sistema operacional é mais fácil que com os SFCs do S7.

Copiar

Usado para copiar estruturas de dados complexas — vetores de instâncias de tipos de dados do usuário.

S7	Comentário	Logix	Comentário
SFC20 BLKMOV	Com o BLKMOV, os endereços devem ser definidos no momento da compilação.	COP (instrução)	Se COP for usado para copiar entre vetores, o início do bloco (origem ou destino) pode incluir um índice de vetor para lidar com o elemento cujo valor é avaliado no momento da execução.
SFC81 UBLKMOV	Versão sem interrupção – para assegurar que os dados de origem não possam ser alterados durante a cópia.	CPS (instrução)	Versão sem interrupção – para assegurar que os dados de origem não possam ser alterados durante a cópia.
SFC14 DPRD_DAT	Se o dispositivo DP Profibus tiver a área de dados de comunicação > 4 bytes, o SFC assegurará leituras consistentes.	CPS (ControlNet e Ethernet /IP)	Não obrigatório para DeviceNet
SFC15 DPWR_DAT	Se o dispositivo DP Profibus tiver área de dados de comunicação > 4 bytes, o SFC assegurará gravações consistentes.	CPS (ControlNet e Ethernet /IP)	Não obrigatório para DeviceNet

Configuração e leitura de data e hora

Os controladores de ambos os sistemas têm um relógio em tempo real que pode ser lido ou configurado.

S7	Comentário	Logix	Comentário
SFC0 SET_CLK	Os valores passados em uma instância do tipo DT (DateTime)	SSV (Set System Value)	Classe SSV - WallClockTime Atributo SSV - DateTime Origem SSV - especificar elemento[0] de DINT[7]
SFC1 READ_CLK	Valores retornados em uma instância do tipo DT (DateTime)	GSV (Get System Value)	Classe GSV - WallClockTime Atributo GSV - DateTime Destino GSV – elemento[0] de DINT[7]

Leitura de hora do sistema

Os controladores de ambos os sistemas têm um relógio de sistema, que dispara junto com o controlador. No sistema S7, o tempo está em milissegundos, no Logix está em microssegundos.

S7	Comentário	Logix	Comentário
SFC64 TIME_TCK	Retorna a hora do sistema na faixa 0…2.31 ms	GSV (Get System Value)	Retorna a hora do sistema no intervalo 0…2.63 µs Classe GSV - CST Atributo GSV - CurrentValue Destino GSV - especifica o elemento[0] de DINT[2] DINT[0] - inferior 32 bits DINT[1] - superior 32 bits

Tratamento de interrupções

As interrupções podem ser habilitadas e desabilitadas pela chamada a funções do sistema pelo usuário do programa.

S7	Comentário	Logix	Comentário
SFC39 DIS_IRT	Desabilita interrupções tratadas por um OB especificado. As requisições de interrupção são perdidas.	SSV Inibe a tarefa especificada.	Classe SSV - Task Instância SSV - Task name Atributo SSV - InhibitTask Origem SSV - variável DINT definido como 1
SFC39 EN_IRT	Habilita interrupções tratadas por um OB especificado	SSV Habilita a tarefa especificada.	Classe SSV - Task Instância SSV - Task name Atributo SSV - InhibitTask Origem SSV - variável DINT definida como 0
SFC41 DIS_AIRT	Desabilita interrupções tratadas por um OB especificado. As requisições de interrupção são atrasadas.	UID	Desabilita a interrupção da tarefa atual por uma tarefa de prioridade mais alta
SFC42 EN_AIRT	Habilita interrupções tratadas por um OB especificado. Quaisquer interrupções atrasadas por SFC41 são executadas.	UIE	Habilita interrupções da tarefa atual.

Erros

Estas chamadas do sistema retornam campos de bit no caso do S7 ou um inteiro no caso do Logix, representando códigos de erro.

S7	Comentário	Logix	Comentário
SFC38 READ_ERR	Lê e limpa bits de erro. O tipo de erro a ser consultado por ser selecionado com um campo de filtragem.	GSV (Use SSV para reinicializar contadores ou falhas)	Classe GSV - FaultLog Atributo GSV: MajorEvents – nenhum evento principal MinorEvents – nenhum evento secundário MajorFaultBits – falha principal atual MinorFaultBits – falha secundária atual Destino GSV – INT ou DINT a receber dados

Status – controlador

A chamada ao SFC (S7) e GSV (Logix) retornará dados sobre o controlador. Observação – o SFC51 requer algum aprendizado antes de poder ser usado. O GSV neste caso é mais acessível.

S7	Comentário	Logix	Comentário
SFC51 RDSYSST	Os parâmetros de entrada especificam a classe de informações a serem lidas e possivelmente um número de instância se houver diversos objetos. Os parâmetros de saída são um indicador para uma lista com as informações retornadas e o número e tamanho dos elementos na lista.	GSV	Módulos com uma conexão direta: Examina o membro "Fault" ou "ChannelFault" se presente. Módulos com uma conexão otimizada de rack: examina o membro "SlotStatusBits" dos dados de entrada do adaptador ou o membro "Fault" do cartão conforme as instruções acima. Para todos os outros cartões: Executa GSV: Classe – Module Instância – ModuleName Atributo - Entrystatus

Status – módulo

A chamada de SFC (S7) e GSV (Logix) retornará dados nos módulos instalados.

S7	Comentário	Logix	Comentário
SFC51 RDSYSST	Os parâmetros de entrada especificam a classe de informações a serem lidas e possivelmente um número de instância se houver diversos objetos. Os parâmetros de saída são um indicador para uma lista com as informações retornadas e o número e tamanho dos elementos na lista.	GSV	Classe GSV - Module Atributo GSV: EntryStatus (relacionamento do objeto Module com o módulo) FaultCode FaultInfo ForceStatus LEDStatus Modo (SSV também) Destino GSV – depende do atributo escolhido

Você pode monitorar as informações de falha nos tags do Logix criados quando o módulo é inserido na configuração de E/S. Da mesma maneira que no STEP 7, se você acessar a configuração de hardware e alternar para "Abrir ONLINE", as informações de falha dos módulos serão exibidas.

Status – for OBs e tarefas

S7	Comentário	Logix	Comentário
Cabeçalho OB	Os dados de status de OBs são armazenados nas variáveis temporárias geradas automaticamente pelo cabeçalho de OB. Eles podem ser acessados diretamente pelo código OB e transferidos às áreas de dados estáticos se o acesso for solicitado de fora do OB. Veja um exemplo abaixo.	GSV/SSV	Classe GSV - Task Instância GSV – Task name Atributo GSV: DisableUpdateOutputs (no fim da Task) EnableTimeOut InhibitTask Instância LastScanTime (microssegundos) MaxIntervaln (entre execuções sucessivas de Task) OverlapCount (disparado na execução) Prioridade Taxa (período em microssegundos) StartTime (valor de WallClockTime quando a tarefa foi iniciada na última vez) Status (3 bits de status) Watchdog (microssegundos) Origem/destino GSV – depende dos atributos escolhidos

Temporizadores

S7	Comentário	Logix	Comentário
SFB4 TON	Temporizador com atraso na energização	TON (LD) TONR (ST & FBD)	Temporizador com atraso na energização
		RTO (LD) RTOR (LD & ST)	Retenção no temporizador com atraso na energização
SFB5 TOF	Temporizador com atraso na desenergização	TOF (LD) TOFR (ST & FBD)	Temporizador com atraso na desenergização
SFB3 TP	Gera um pulso executado incondicionalmente	Bit do acumulador de execução livre	

Rotinas de conversão

S7	Comentário	Logix	Comentário
Funções de biblioteca		**Instruções**	
FC16 I_STRNG	Número inteiro para o grupo	DTOS	INT pode ser usado como um tag de origem em vez de DINT
FC5 DI_STRNG	Número inteiro duplo para o grupo	DTOS	DINT para o grupo
FC30 R_STRG	Real para o grupo	RTOS	Real para o grupo
FC38 STRG_I	Grupo para o número inteiro	DTOS	
FC37 STRG_DI	Grupo para o número inteiro duplo	STOD	Grupo para DINT
FC39 STRG_R	Grupo para real	STOR	Grupo para real

Rotinas de manuseio de grupo

S7	Comentário	Logix	Comentário
	Funções de biblioteca		Instruções
FC10 EQ_STRNG	Compara grupos quando à igualdade	EQU	Compara grupos quando à igualdade
FC13 GE_STRNG	Compara grupos quanto a >=	GEQ (LD) >= (ST)	Compara grupos quanto a >=
FC15 GT_STRNG	Compara grupos quanto a >	GRT (LD)	Compara grupos quanto a >
FC19 LE_STRNG	Compara grupos quanto a <=	LEQ (LD) <= (ST)	Compara grupos quanto a <=
FC24 LT_STRNG	Comparar grupos quanto a <	LES (LD) < (ST)	Comparar grupos quanto a <
FC29 NE_STRNG	Compara grupos quanto a <>	NEQ (LD) <> (ST)	Compara grupos quanto a <>
FC21 LEN	Comprimento do grupo	.LEN	Propriedade de qualquer instância do grupo
FC26 MID	Retorna uma seção intermediária do grupo	MID	Retorna uma seção intermediária do grupo
FC2 CONCAT	Concatena dois grupos	CONCAT	Concatena dois grupos
	Pode ser feito com FC31 REPLACE	DELETE	Exclui uma seção de um grupo
FC17 INSERT	Insere o grupo de origem no grupo de destino	INSERT	Insere o grupo de origem no grupo de destino
FC31 REPLACE	Substitui n caracteres do grupo de destino pelo grupo de origem	Usa DELETE/INSERT	
FC11 FIND	Localiza um grupo em outro grupo	FIND	Localiza um grupo em outro grupo

Não há equivalente no STEP 7 para as instruções de porta serial ASCII do Logix – nem no conjunto de instruções nem na biblioteca de funções. Essas terão de ser programadas em STL se necessário.

Exemplos de chamadas de função do sistema

Esses exemplos destinam-se principalmente a ilustrar o uso de instruções GSV/SSV.

Ajustando o relógio

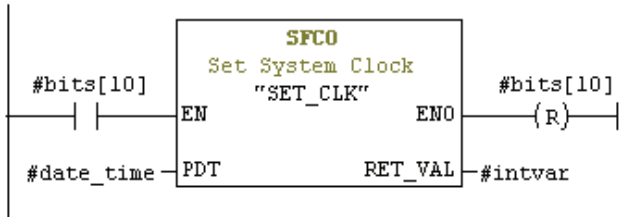
STEP 7

Esta chamada ao SFC0 ajustará o relógio. A hora e data são inseridas em #date_time.

A data e hora são armazenadas em 8 bytes seguindo o formato #date_time em BCD.

Network 14 : Title:

```
set the clock to the value stored in "date_time"
```



0 – ano

1 – mês

2 – dia

3 – hora

4 – minuto

5 – segundo

6 – 2 dígitos mais significativos de milissegundos

7 – 1 dígito menos significativo de milissegundos e dia da semana

Logix

Os valores de data e hora são armazenados nos sete DINTs seguindo #date_time.

0 – ano

1 – mês

2 – dia

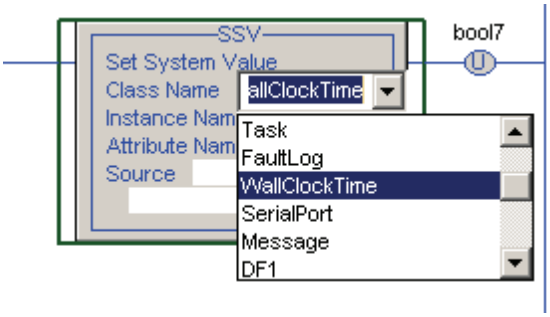
3 – hora

6 – minuto

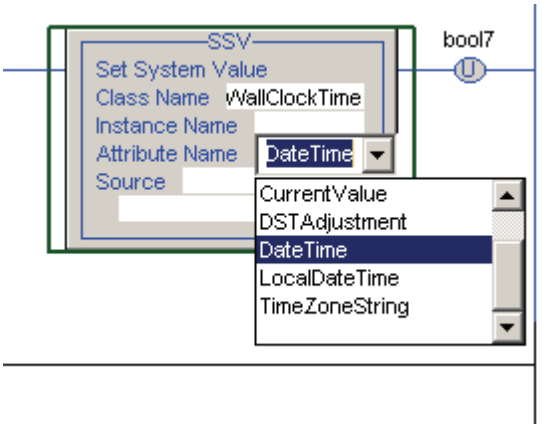
5 – segundo

6 – microssegundo

A captura de tela do Logix mostra a estrutura de dados associada a GSV e SSV. Selecione a classe de um menu como apresentado.



Selecione Atributo no menu, como apresentado.



Finalmente, selecione o tag que será a origem (SSV) ou destino (GSV) dos dados.

Desabilitando interrupções

STEP 7

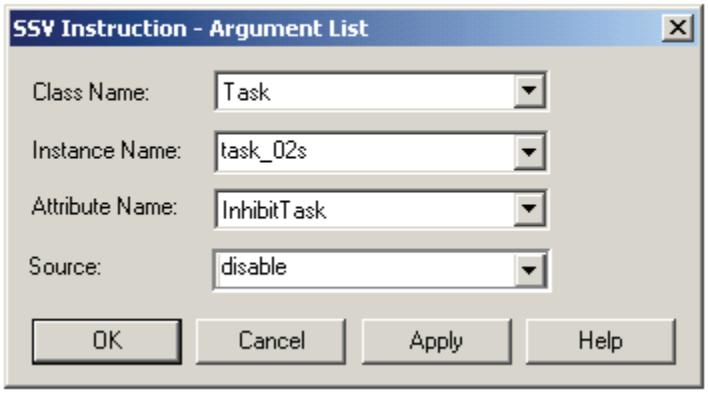
```
Network 2 : Title:
Disable interrupts for the Interrupt Execution (ie Periodic) OB35

          CALL  "DIS_IRT"                SFC39          -- Disable New Inte
           MODE   :=B#16#2
           OB_NR  :=35
           RET VAL:=#intVar
```

Logix

Este exemplo mostra SSV em texto estruturado.

Se você digitar “gsv”, depois “alt-A”, a tela de seleção de parâmetro a seguir será exibida.

```
// disable task_0.2s
ssv()
```

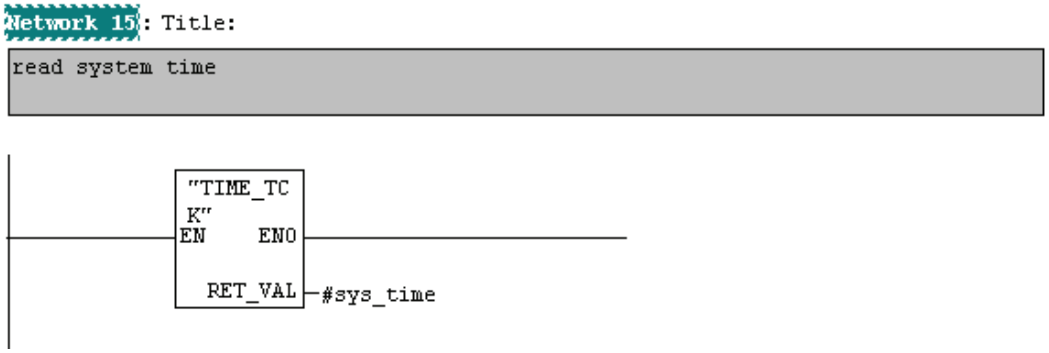


Após inserir os parâmetros, clique em “OK” e os parâmetros reais serão completados.

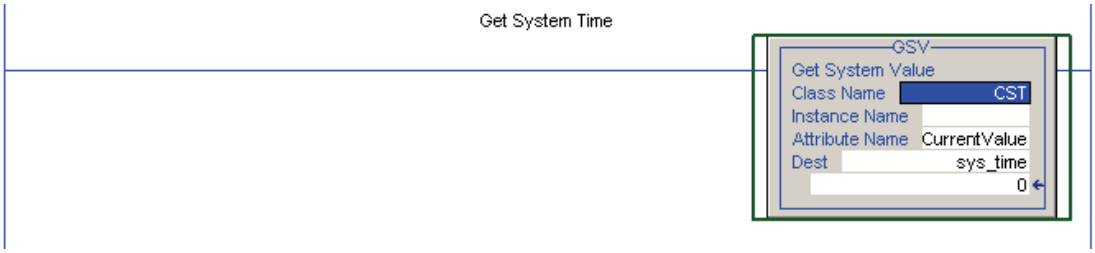
```
// disable task_0.2s
ssv(Task,task_02s,InhibitTask,disable);
```

Ler hora do sistema

STEP 7



Logix

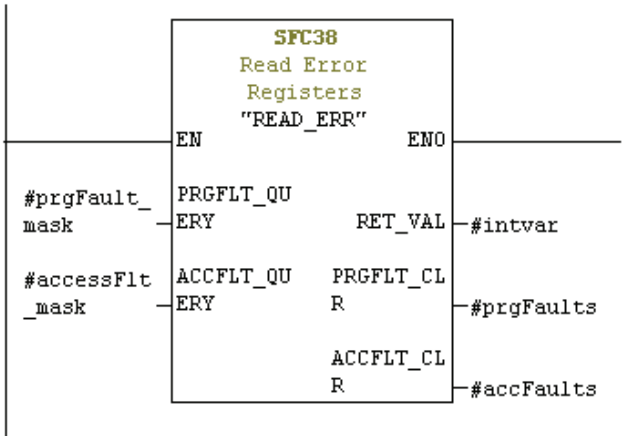


Obter falhas

STEP 7

Network 16 : Title:

```
Get programming faults and I/O access faults
```



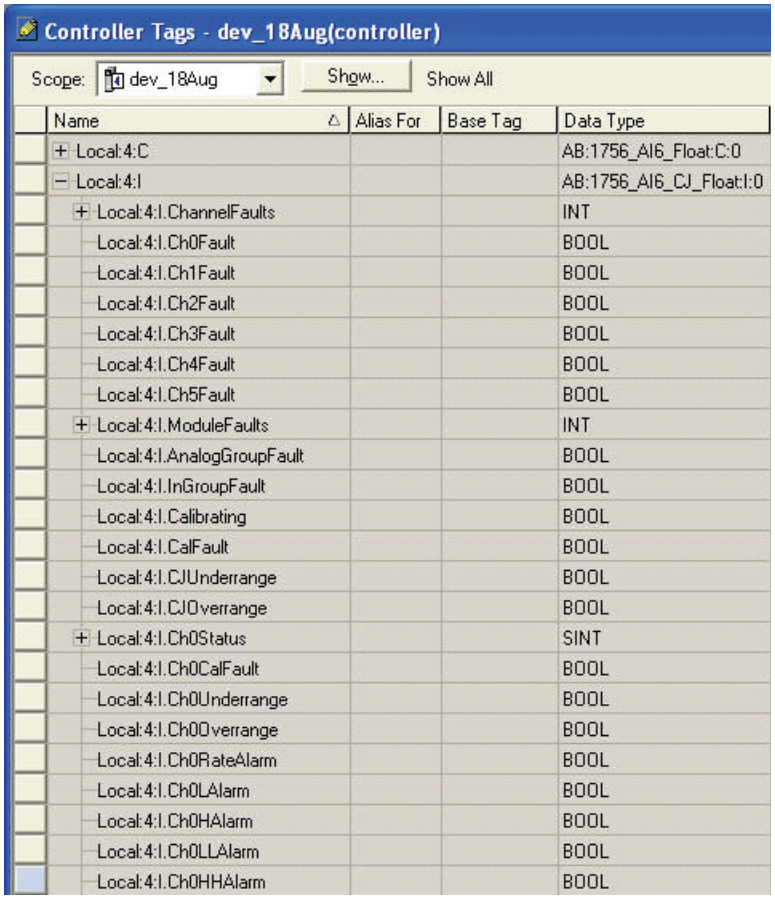
O padrão de bits nos parâmetros de entrada agem como um filtro para selecionar as falhas a serem consultadas. As falhas retornadas são as falhas **mascaradas** – o mascaramento evita a parada do controlador ou chamada de um OB com falha.

Logix

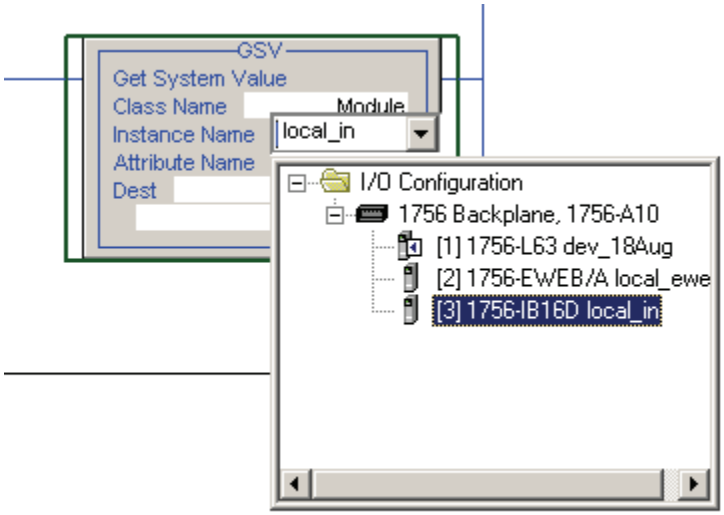
Informações do módulo

A maneira mais fácil de inspecionar os tags do perfil do dispositivo do módulo, que contêm informações de falha/diagnóstico.

1756-IT6I2 Tag de entrada analógica de termopar



Outra maneira de usar as instruções GSV para ler objetos do módulo. A captura de tela abaixo mostra como usar o GSV par obter informações a respeito do módulo de entrada digital 1756-IB16D.



Obter tempo de varredura

STEP 7

Esta é uma captura de tela do cabeçalho Variáveis Temporárias de OB1.

Contents Of: 'Environment\Interface\TEMP'

Name	Data Type	Address	Comment
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And...	12.0	Date and time OB1 started

#OB1_PREV_CYCLE é o tempo de varredura. Como uma variável temporária, ela deixará de existir quando a execução de OB1 for concluída. Para armazenar o tempo de varredura, copie #OB1_PREV_CYCLE para um local de memória estática.

Logix

O tempo de execução pode ser recuperado de cada tarefa do Logix.

Com o S7, você pode obter diretamente o tempo de varredura para OB1 de #OB1_PREV_CYCLE. Entretanto, para OBs periódicos, não há equivalente para #OB1_PREV_CYCLE. Para obter o tempo de execução de OBs periódicos, você precisará inserir chamadas para SFC64 TIME_TCK no início e fim do OB e subtrair os tempos do relógio do sistema retornados pelo SFC.

Conversão de estruturas de programa típicas

Introdução

O objetivo desta seção é demonstrar como algumas tarefas de programação típicas no STEP 7 podem ser executadas no software RSLogix 5000. A explicação baseia-se principalmente em fragmentos de código, mas também há alguns exemplos complexos.

Tópico	Página
Exemplos de código de conversão	83
Outros tópicos relacionados à programação	120
Um exemplo maior - módulo de controle	121

Há também alguma explicação sobre os problemas relacionados à programação, como o escopo e visibilidade de variáveis e a programação de seções de código.

Exemplos de código de conversão

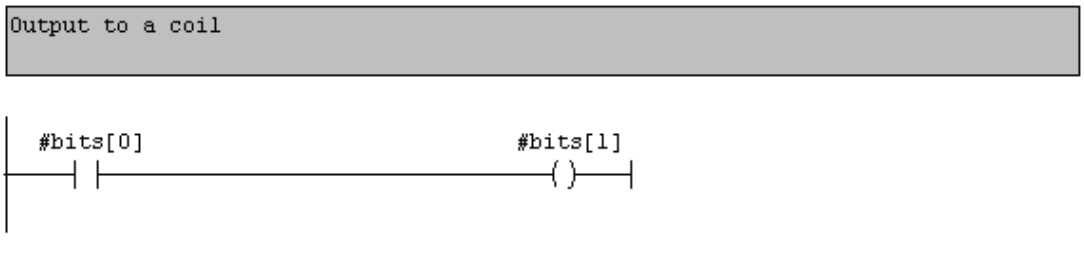
Estes exemplos mostram o código de conversão.

Conversão de lógica ladder

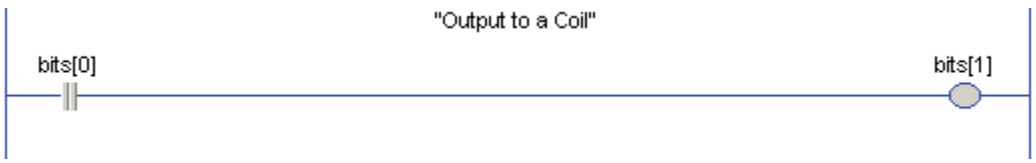
Esta seção descreve alguns exemplos de comparação entre o LAD do STEP 7 e LD do Logix.

Gravação em uma bobina

STEP 7

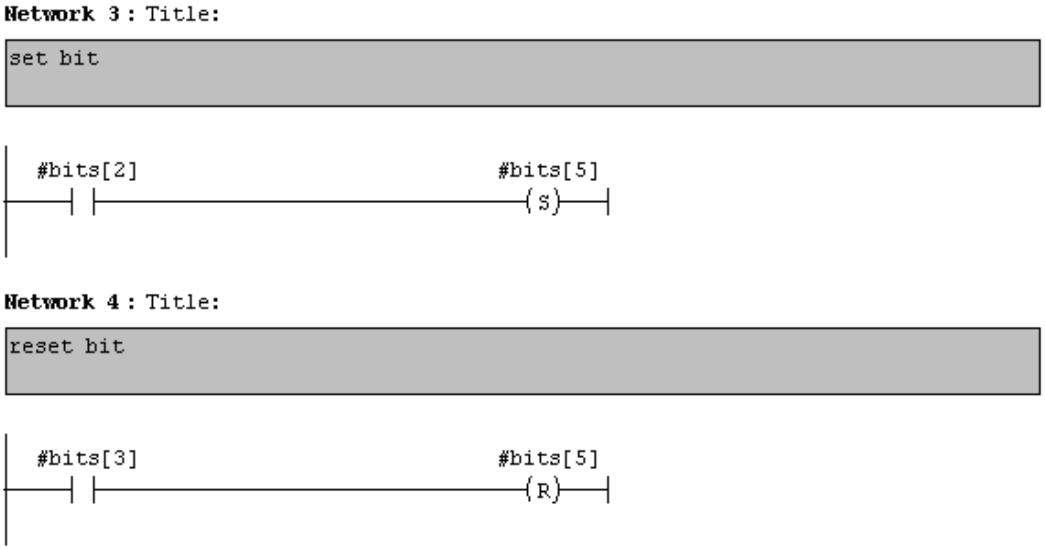


LOGIX



Definir e reinicializar

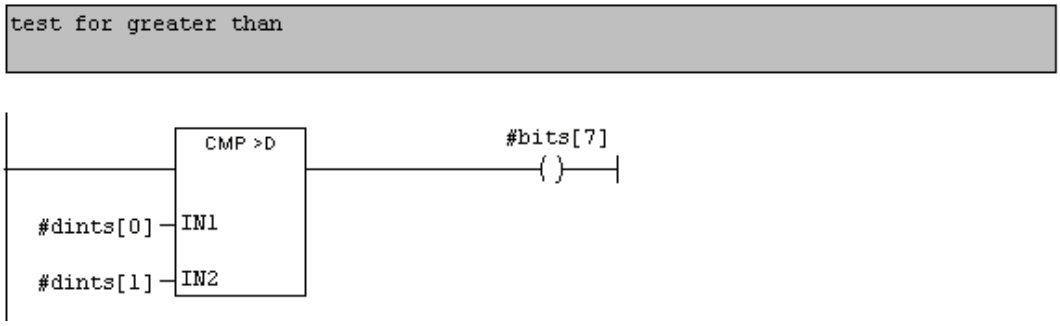
STEP 7



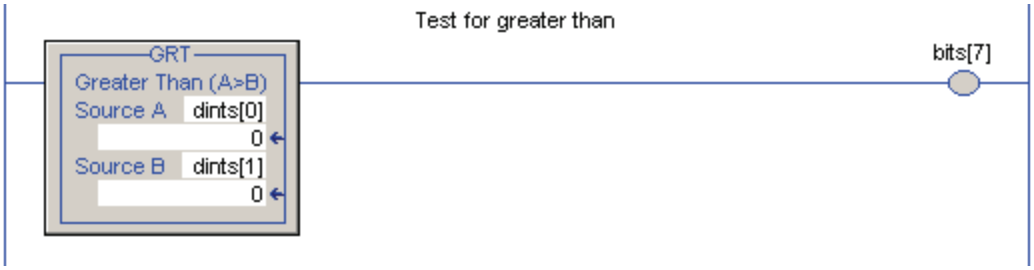
LOGIX

Teste para maior que

STEP 7



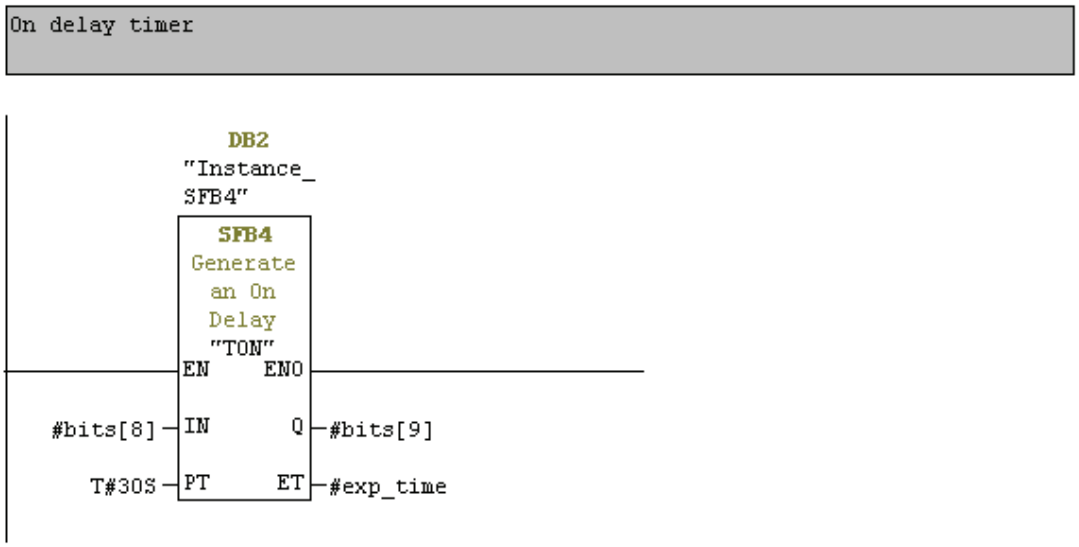
LOGIX



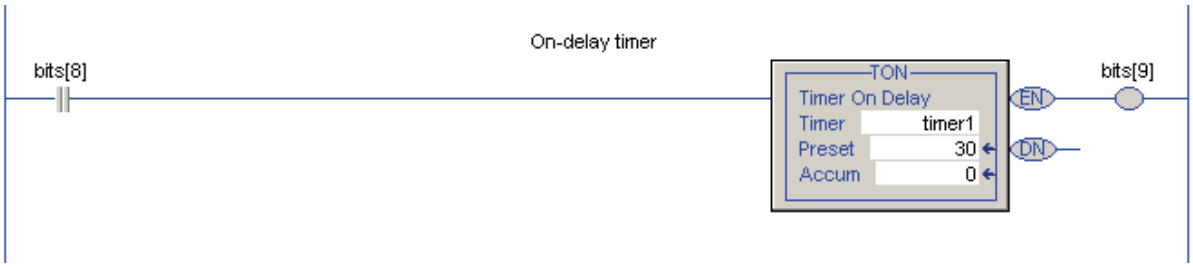
Como antes, use a instrução CMP se a expressão for mais complexa do que apenas comparar dois números.

No atraso do temporizador

STEP 7



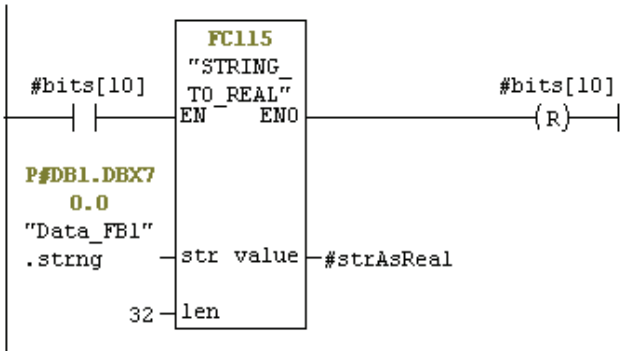
LOGIX



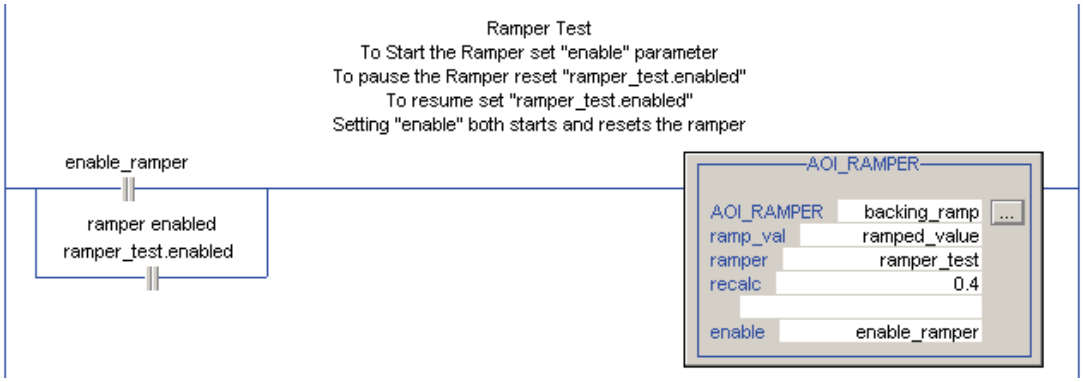
Chamada de função do usuário

STEP 7

```
user-function call
```

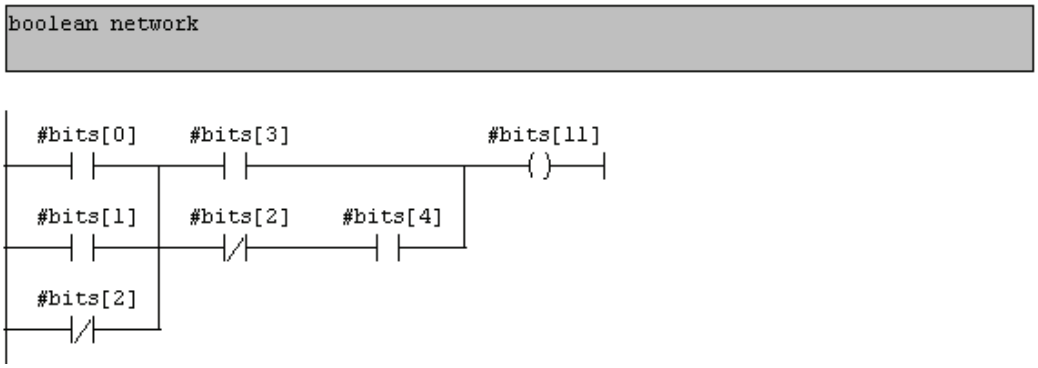


LOGIX

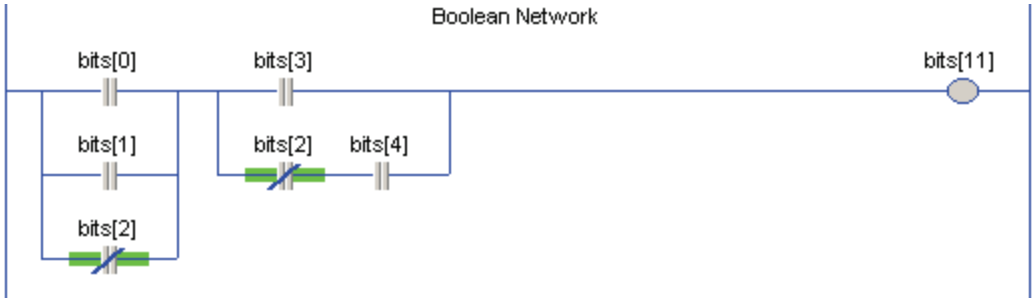


Rede booleana

STEP 7



LOGIX

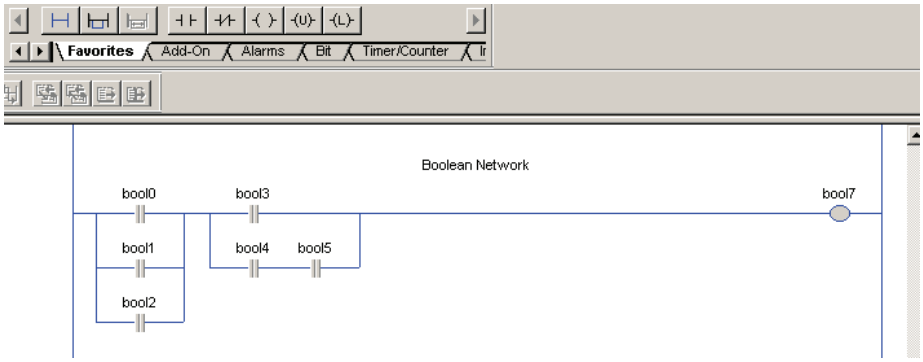


Há semelhança suficiente entre o LAD STEP 7 LAD e o LD Logix LD para tornar a conversão no nível de rotinas bastante direta.

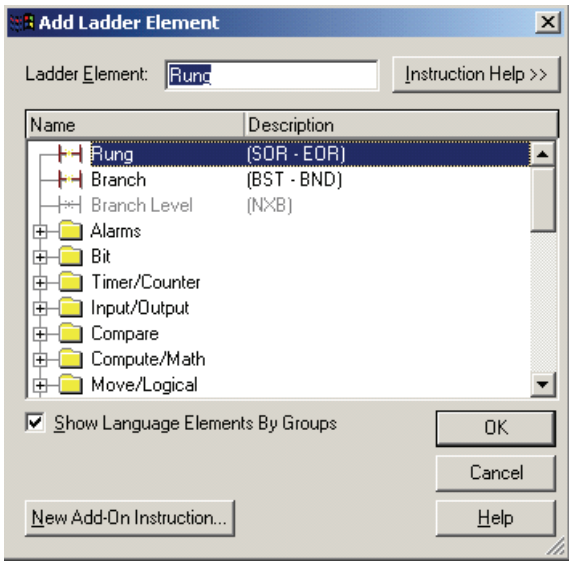
O editor LD do Logix

Há não menos do que sete maneiras de selecionar instruções LD. Dois métodos muitos semelhantes ao modo como isso é feito no STEP 7 são descritos abaixo.

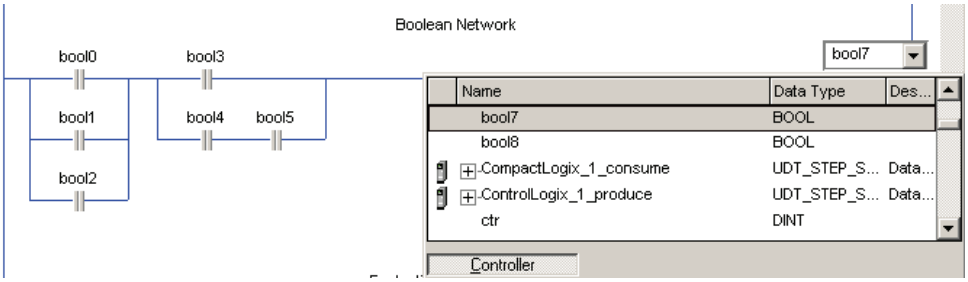
Você pode selecionar de uma paleta acima a folha de dados LD.



Se você digitar Alt+Insert, esse menu pop-up de seleção será exibido.



Ao configurar instruções, menus estarão disponíveis para permitir que você selecione o tag a ser inserido.



Saltos e tomada de decisões

STEP 7 - seqüência de salto convencional

A tarefa de exemplo a seguir é explicada no comentário da rede. Duas versões S7 são mostradas devido a ambas serem usadas com freqüência.

```
Network 1: Multi-way selection
if #input is 5 set #target to 8

else if #input is 6 set #target to 10

else if #input is 7 set #target to 16

else set #input to 0
```

```
      L     #input
      L     5
      ==I
      JCN   _001

      L     8
      T     #target

      JU    end

_001: L     #input
      L     6
      ==I
      JCN   _002

      L     10
      T     #target

      JU    end

_002: L     #input
      L     7
      ==I
      JCN   _003

      L     16
      T     #target

      JU    end

_003: L     0
      T     #target

end:  NOP   0
```

O valor do número da entrada é comparado ao conjunto de constantes até a comparação ser encontrada. Então a comparação é executada e a comparação acaba. Uma ação padrão será executada se #input não se comparar a qualquer valor no conjunto.

STEP 7 - Lista de salto

Neste exemplo, a tarefa é a mesma, mas uma lista de salto é usada.
Isso é semelhante a uma tabela de salto de microprocessador e transfere a execução a um rótulo, dependendo do valor de uma variável.

Network 2 : Title:

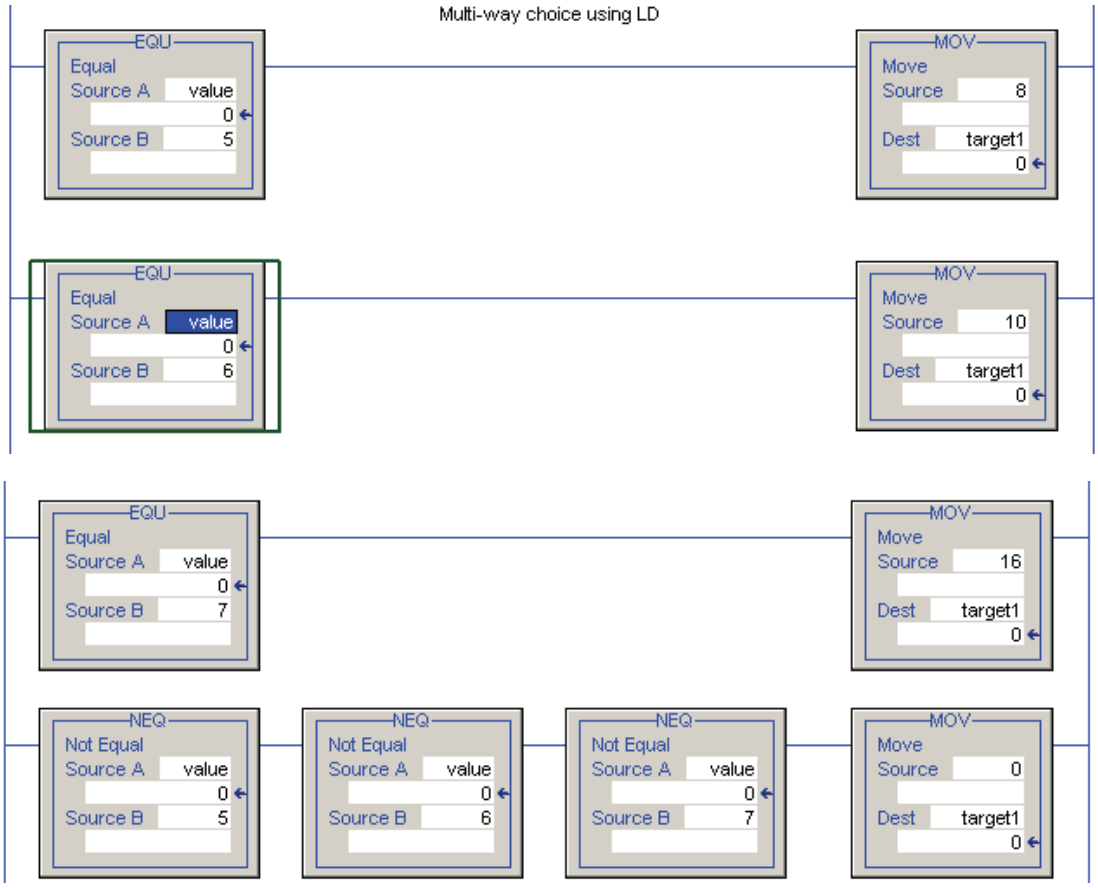
```
if #input is 5 set #target to 8

else if #input is 6 set #target to 10

else if #input is 7 set #target to 16

else set #input to 0
```

```
          L     #input
          L     5
          -I
          JL    rng
          JU    d5
          JU    d6
          JU    d7
    rng:  L     0
          T     #target
          JU    cont

    d5:   L     8
          T     #target
          JU    cont

    d6:   L     10
          T     #target
          JU    cont

    d7:   L     16
          T     #target

    cont: NOP   0
```

Isso é mais fácil de ler do que uma seqüência de salto convencional e é mais eficiente pois apenas o código na tabela de destino é executado.

Logix - lógica ladder

Isso mostra uma escolha múltipla usando LD.



Logix - texto estruturado If. . . Then. . . Else

Qualquer um familiarizado com a linguagem de programação nas famílias Basic/Pascal/C entenderá isso sem dificuldade.

```
// multi-way choice using Structured Text

if (value = 5) then target := 8;
elsif (value = 6) then target := 10;
elsif (value = 7) then target := 16;
else target := 0;
end_if;
```

Os colchetes em volta da condição “if” não são obrigatórios.

Declaração CASE de texto estruturado Logix

Essa é outra variação no ST que faz a mesma tarefa. Ela é tão compacta e limpa que há pouca necessidade de comentários adicionais.

```
// multi-way choice using Structured Text CASE

case value of
    5: target := 8;
    6: target := 10;
    7: target := 16;
else target :=0;
end_case;
```

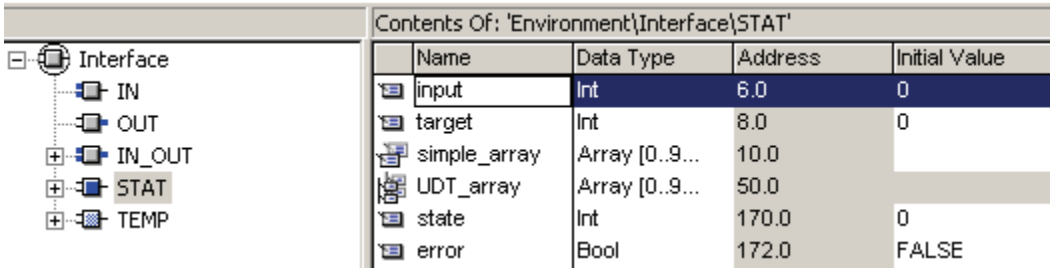
Todas as soluções funcionam, mas esta é a solução Logix preferencial. Ela é tão compacta e limpa que há pouca necessidade de comentários adicionais.

Vetores

Tanto o STEP 7 quanto o Logix permitem que vetores de objetos simples ou complexos sejam criados na memória. O Logix tem suporte de alto nível para acessar vetores. No STEP 7, entretanto, programação de baixo nível é necessária.

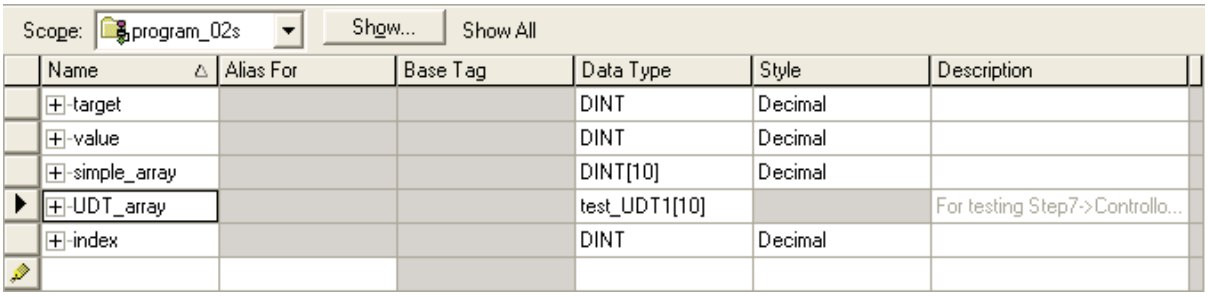
Criação de vetores no STEP 7

A captura de tela a seguir mostra dois vetores criados em uma instância de bloco de dados. Simple_array é um vetor de 10 elementos. UDT_array é um vetor de 10 estruturas do tipo test_UDT1, onde test_UDT1 é um tipo de dados do usuário contendo alguns outros tipos, não mostrados.



Criação de vetores no Logix

Isso é exatamente igual no Logix.



Sintaxe de declaração de vetor

O STEP 7 usa a sintaxe de declaração VETOR[0...15] DE REAL. O Logix usa REAL[15].

O STEP 7 tem uma sintaxe especial para grupos. STRING[32] é um grupo de 32 caracteres no STEP 7 enquanto no Logix GRUPO[32] é um vetor de 32 grupos, cada um contendo 82 caracteres.

Acesso de vetor no STEP 7

Este exemplo é para executar uma tarefa simples em dois vetores simple_array[] e UDT_array[]. A tarefa é descrita nos comentários da rede.

No STEP 7, não é possível acessar vetores usando a notação vetor[] normal. Em vez disso, você deve usar operações de baixo nível com indicadores. No fragmento abaixo, uma função "GET_INDEXED_REFERENCE" torna a tarefa muito mais fácil ao retornar um ponteiro ao elemento de vetor a ser acessado.

```
Network 3 : Title:
array operations
----------------
if (simple_array[2] = simple_array[5]) then
  UDT_array[8].booleanl := 1;
else
  UDT_array[8].booleanl := 0;
end_if;
```

```
    // 1.  compare simple_array[2] with simple_array[5]
          CALL  "GET_INDEXED_REFERENCE"           FC111
           refArray  :="Data_test".simple_array   P#DB1.DBX10.0
           index     :=2
           byteIncr  :=4
           startIndex:=FALSE
           retVal    :=#ptr1                      BOOL

          CALL  "GET_INDEXED_REFERENCE"           FC111
           refArray  :="Data_test".simple_array   P#DB1.DBX10.0
           index     :=5
           byteIncr  :=4
           startIndex:=FALSE
           retVal    :=#ptr2

          OPN   "Data_test"                       DB1
          L     DID [#ptr1]
          L     DID [#ptr2]
          ==D
          =     #compare


   // 2. get pointer to UDT_array
          CALL  "GET_INDEXED_REFERENCE"           FC111
           refArray  :="Data_test".UDT_array      P#DB1.DBX44.0
           index     :=8
           byteIncr  :=12     INT
           startIndex:=FALSE
           retVal    :=#ptr1

          L     #ptr1
          LAR1

   // 3. set or reset the bit
          A     #compare
          =     DIX [AR1,P#0.0]
```

Nesse caso, o código de texto estruturado Logix foi usado como o comentário da rede, demonstrando quão intuitivo o código Logix é.

STEP 7 - Criando loops nos elementos de vetor

O objetivo neste exemplo é limpar o campo flutuante em cada estrutura em UDT_array[]. Isso não é difícil, mas a confiança no uso de indicadores é claramente necessária.

```
Array Operations
----------------
Clear all float elements at UDT offset P#6.0 in array UDT_array
```

```
// transfer pointer to UDT_array to AR1
      L     P##UDT_array
      LAR1
// initialise counter
      L     0
      T     #ctr

// end if #ctr > 9
loop: L     #ctr
      L     9
      >I
      JC    end2
// clear the float field at offset p#6.0
      L     0.000000e+000
      T     DID [AR1,P#6.0]
// increment AR1 by size of the UDT
      +AR1  P#12.0
// increment counter
      L     #ctr
      INC   1
      T     #ctr
// loop back
      JU    loop

end2: NOP   0
```

Logix - operações de vetor em texto estruturado

O fragmento ST a seguir executa as tarefas descritas nas duas seções precedentes.

```
// array access in ST
if (simple_array[2] = simple_array[5]) then
    UDT_array[8].booleanl := 1;
else
    UDT_array[8].booleanl := 0;
end_if;


// clearing array elements
if (simple_array[0] = 5) then
    index := 0;
    while (index <= 9) do
        UDT_array[index].reall := 0.0;
        index := index + 1;
    end_while;
end_if;
```

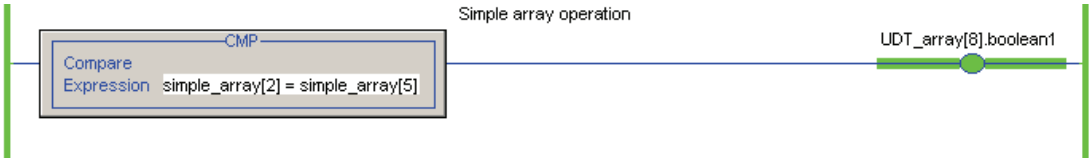
Nenhum comentário adicional é necessário para descrever como isso funciona.

Se precisar ficar alternando em variáveis booleanas com declarações if...then...else, considere escrever uma equação booleana em vez disso:

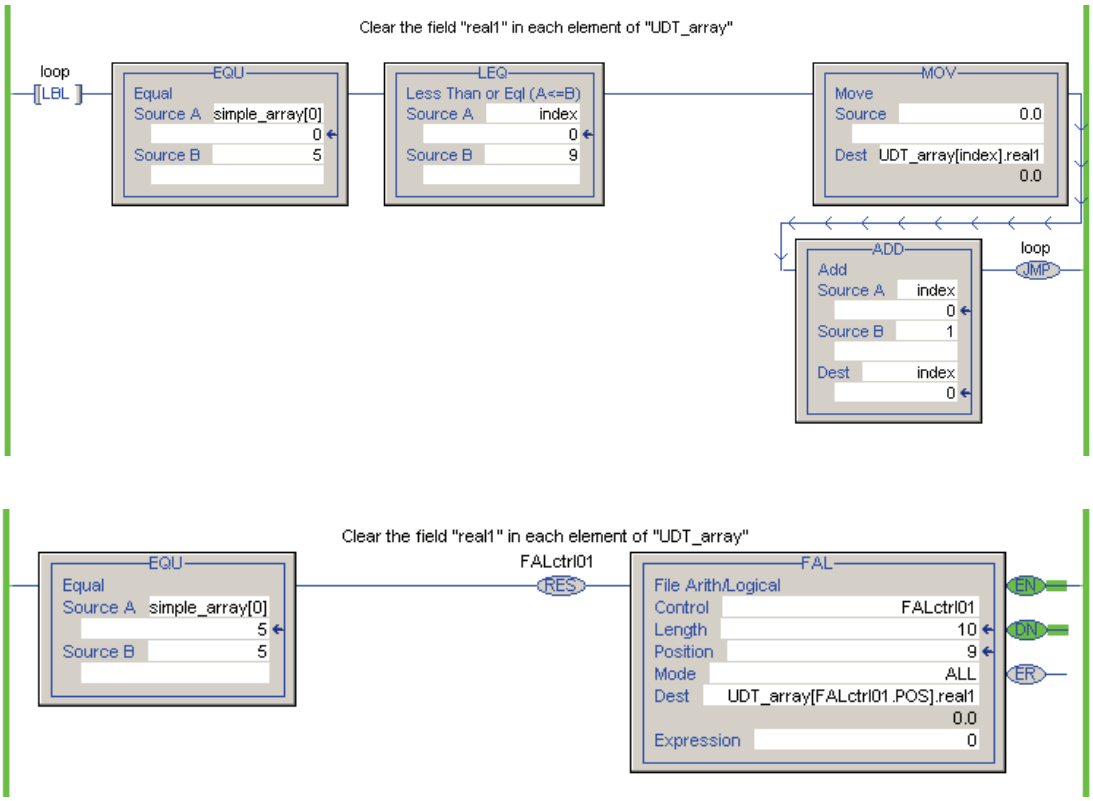
```
// array access in ST
UDT_array[8].booleanl := simple_array[2] = simple_array[5];
```

Logix - operações de vetor no diagrama ladder

Os exemplos da seção anterior podem ser escritos em LD usando a instrução CMP (comparar) como mostrado a seguir.



O segundo (limpando o campo real no vetor de UDTs) pode ser feito de uma destas maneiras.

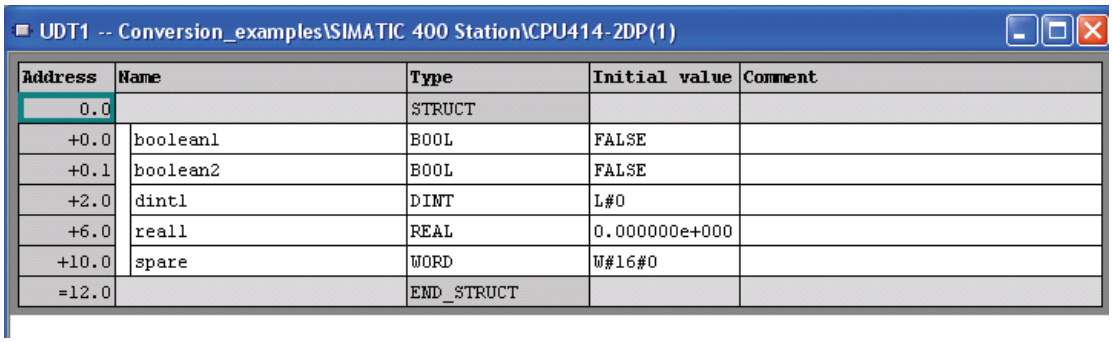


A primeira abordagem para limpar os elementos de vetor é uma conversão do loop While do código ST. O segundo usa a instrução FAL avançada para operações de vetor.

Tipos de dados do usuário

Configurar e usar Tipos de Dados do Usuário (UDTs) no STEP 7 e Logix é muito semelhante.

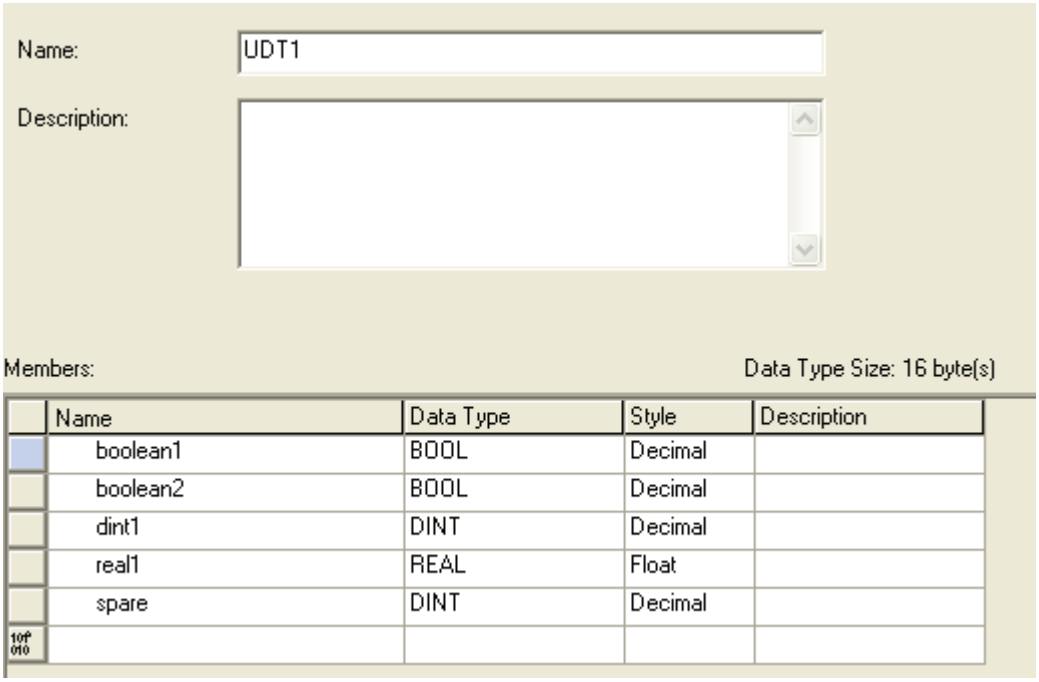
Abaixo está um UDT no STEP 7.



UDT1 -- Conversion_examples\SIMATIC 400 Station\CPU414-2DP(1)

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	boolean1	BOOL	FALSE	
+0.1	boolean2	BOOL	FALSE	
+2.0	dint1	DINT	L#0	
+6.0	real1	REAL	0.000000e+000	
+10.0	spare	WORD	W#16#0	
=12.0		END_STRUCT		

Abaixo está um UDT no Logix.



Name: UDT1

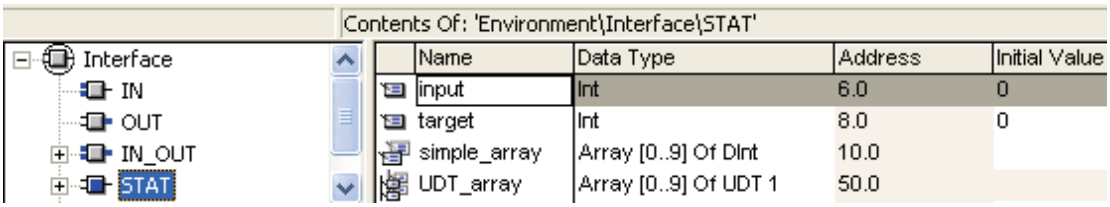
Description:

Members: Data Type Size: 16 byte(s)

	Name	Data Type	Style	Description
	boolean1	BOOL	Decimal	
	boolean2	BOOL	Decimal	
	dint1	DINT	Decimal	
	real1	REAL	Float	
	spare	DINT	Decimal	

Em ambos os sistemas, os UDTs podem ser usados para declarar e definir variáveis.

Eis uma declaração envolvendo um UDT no STEP 7.

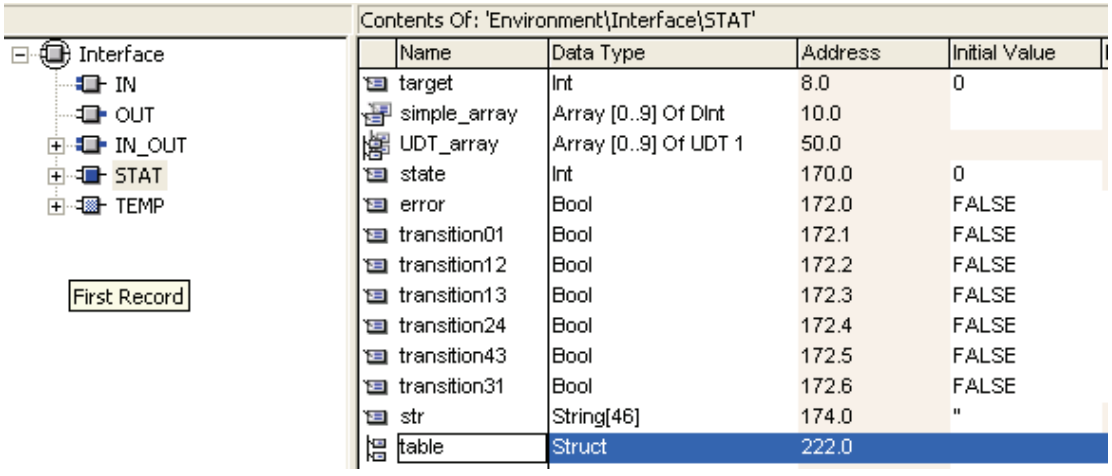


Eis uma declaração envolvendo um UDT no Logix.

Name	Alias For	Base Tag	Data Type	Style
Limit_Switch_1	Local:3:I.Data.0	Local:3:I.Data.0	BOOL	Decimal
Local:3:C			AB:1756_DI:C:0	
Local:3:I			AB:1756_DI:I:0	
conveyor_1			UDT1	
conveyor_1.boolean1			BOOL	Decimal
conveyor_1.boolean2			BOOL	Decimal
conveyor_1.dint1			DINT	Decimal
conveyor_1.real1			REAL	Float
conveyor_1.spare			DINT	Decimal

Uma pequena diferença entre os dois sistemas é a seguinte:

No STEP 7 você pode declarar um tipo de variável “struct”.



Observe a entrada “table” do tipo Struct. Dentro de “table” pode estar uma coleção (ordenada ou desordenada) de qualquer combinação de tipos.

No Logix, isso pode ser feito pela definição de “Struct” como um UDT contendo a estrutura de dados desejada e, então, declarando “tabel” como tipo Struct.

Name	Alias For	Base Tag	Data Type	Style
Limit_Switch_1	Local:3:I.Data.0	Local:3:I.Data.0	BOOL	Decimal
⊞ Local:3:C			AB:1756_DI:C:0	
⊞ Local:3:I			AB:1756_DI:I:0	
⊟ conveyor_1			UDT1	
conveyor_1.boolean1			BOOL	Decimal
conveyor_1.boolean2			BOOL	Decimal
⊞ conveyor_1.dint1			DINT	Decimal
conveyor_1.real1			REAL	Float
⊞ conveyor_1.spare			DINT	Decimal
⊟ table			Struct	
table.status_bit1			BOOL	Decimal
table.status_bit2			BOOL	Decimal
table.status_bit3			BOOL	Decimal
⊞ table.dwell_timer1			TIMER	
⊞ table.dwell_timer2			TIMER	
table.speed			REAL	Float

Ponteiros e vetores

Um programa STEP 7 pode ter indicadores para qualquer objeto de dados. O acesso indireto a blocos de dados é também permitido, mas não há indicadores para funções (exceto de uma maneira restrita pela instrução JL (lista de salto). O ponteiro de dados é incomum no tocante a ser um indicador para um bit. Portanto, seu valor é oito vezes o de um indicador normal para um byte. Isso reflete a importância de bits na programação de sistemas de controle.

No Logix não há indicadores. Os vetores executam a mesma função que os indicadores, mas são mais simples e seguros.

O programador S7 será capaz de executar um conjunto completo de tarefas no Logix sem os indicadores?Na programação de computadores, os indicadores para dados são usados principalmente para três finalidades:

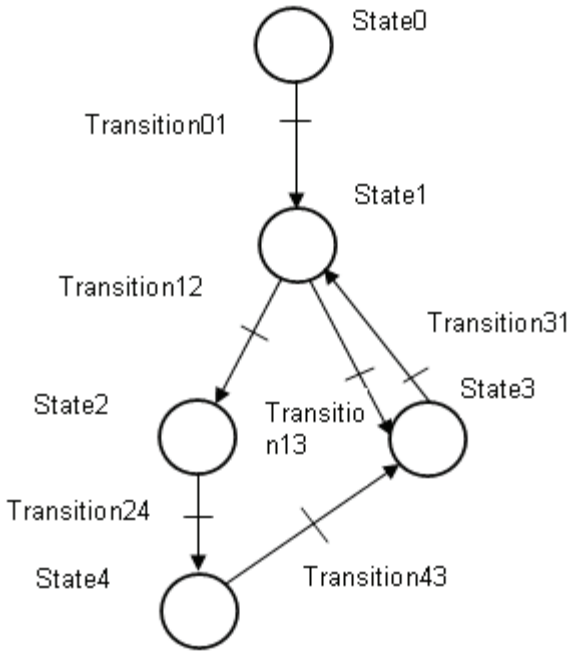
- Operações em itens de dados ordenados seqüencialmente (vetores de objetos, grupos).
- Alocar, acessar e excluir objetos alocados dinamicamente.
- Passar referências a objetos como parâmetros em chamadas de função.

No Logix, a primeira finalidade é atendida pelos vetores. A segunda finalidade não é relevante no controle de software, pois não alocamos objetos dinamicamente. A terceira é atendida pelos parâmetros “inout” tanto nos blocos de funções do STEP 7 quanto nas instruções add-on do Logix.

Conclui-se, portanto, que a ausência de indicadores explícitos não é uma limitação para os programadores em Logix. Os programadores em STEP 7 irão descobrir também que a codificação usando vetores pode ser feita mais rapidamente no texto estruturado usando vetores que no STL usando indicadores.

Máquina de estados

A máquina de estados é um elemento importante no software de sistemas de controle, pois simplifica enormemente a tarefa de controle seqüencial de programação.



Máquina de estados do STEP 7

O STEP 7 oferece controle gráfico seqüencial de funções como uma opção extra à aplicação básica. Se o SFC gráfico não estiver disponível, a lista de declarações fará o trabalho.

```
Network 4: Title:
state machine

      L     #state
      JL    rng1
      JU    st0
      JU    st1
      JU    st2
rng1: SET
      S     #error
      BEU

st0:  L     1
      A     #transition01
      JC    next
      JU    ovr

st1:  L     2
      A     #transition12
      JC    next

      L     3
      A     #transition13
      JC    next
      JU    ovr

st2:  L     4
      A     #transition24
      JC    next
      JU    ovr

st3:  L     0
      A     #transition31
      JC    next
      JU    ovr

st4:  L     3
      A     #transition43
      JC    next
      JU    ovr

next: T     #state

ovr:  NOP   0
```

A variável #state contém o número de estado. A instrução de lista de salto faz com que a execução salte para o rótulo relevante ao valor #state. Se a condição de transição de estado for Verdadeira, o novo valor de estado será carregado no acumulador e a execução saltará para o rótulo label “next”, onde o novo número de estado será transferido para a variável #state.

Máquina de estados Logix em texto estruturado

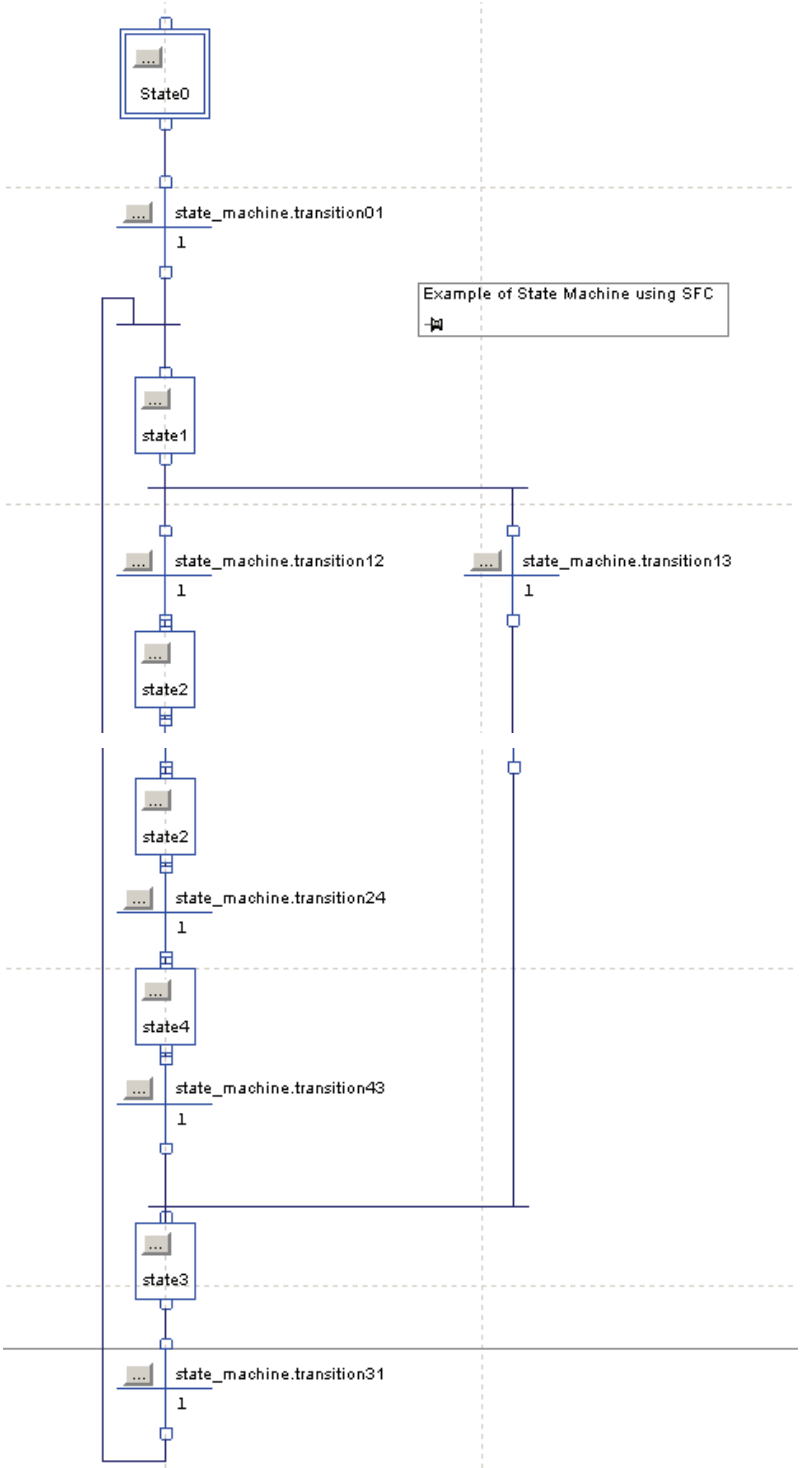
Eis a mesma máquina de estados em texto estruturado, usando a declaração CASE. Como em outros exemplos ST, seria difícil escrever uma descrição mais clara que o código em si.

```
// implementation of State Machine using CASE in ST

case state_machine.state of
    0:  if state_machine.transition01 then
            state_machine.state := 1;
        end_if;
    1:  if state_machine.transition12 then
            state_machine.state := 2;
        elsif state_machine.transition13 then
            state_machine.state := 3;
        end_if;
    2:  if state_machine.transition24 then
            state_machine.state := 4;
        end_if;
    3:  if state_machine.transition31 then
            state_machine.state := 1;
        end_if;
    4:  if state_machine.transition43 then
            state_machine.state := 3;
        end_if;
end_case;
```

Máquina de estados Logix no controle seqüencial de funções

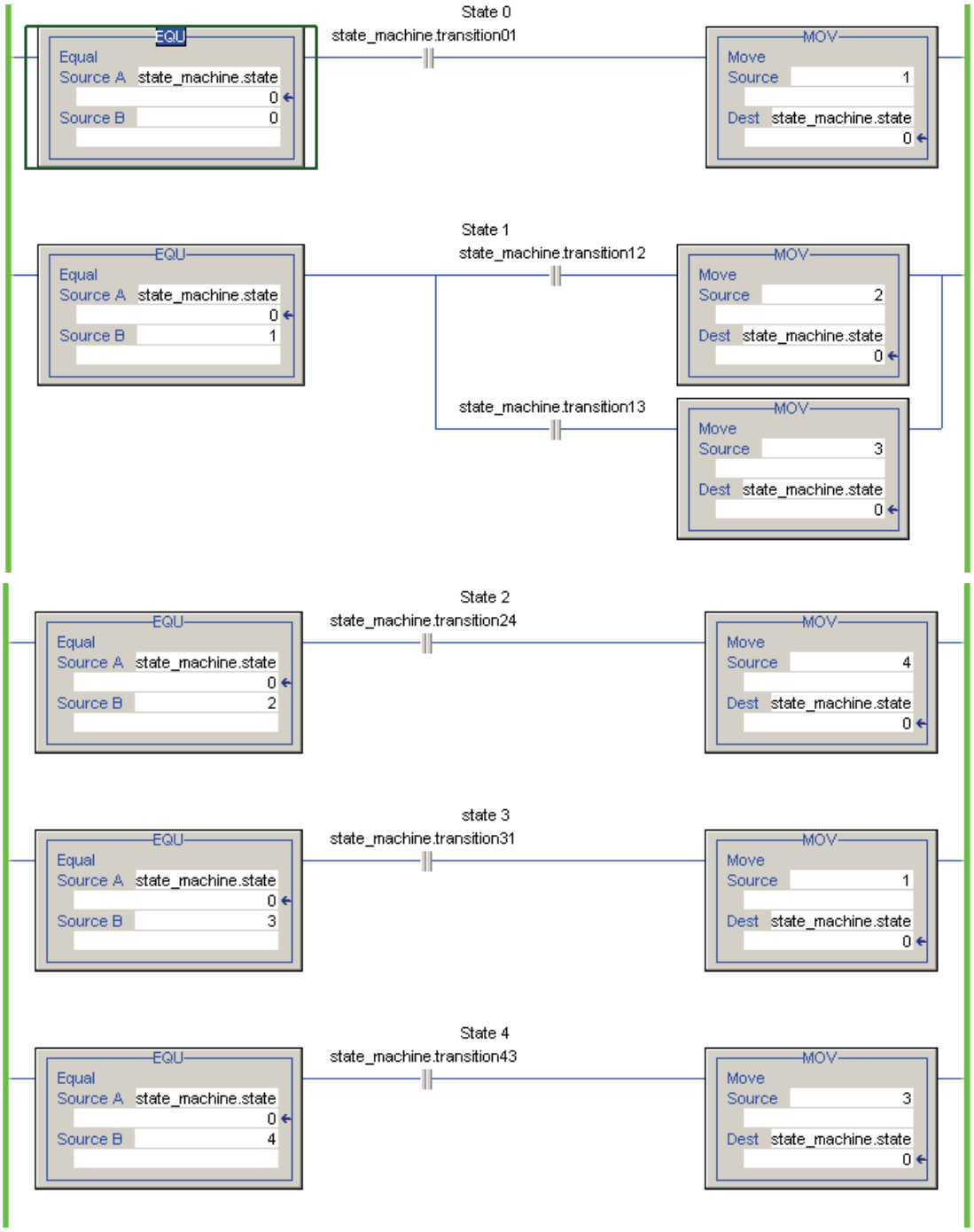
O Logix fornece um SFC gráfico como uma das linguagens de sua suíte padrão. Abaixo está a máquina de estados em SFC.

Implementação da máquina de estados usando SFC



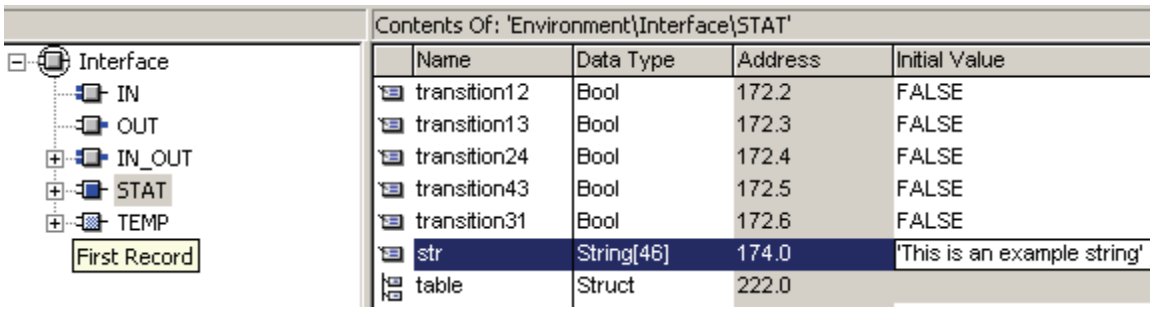
Máquina de estados em diagrama de lógica ladder

A captura de tela abaixo mostra como a máquina de estados pode ser implementada no LD.



Grupos

Definição de grupo no STEP 7



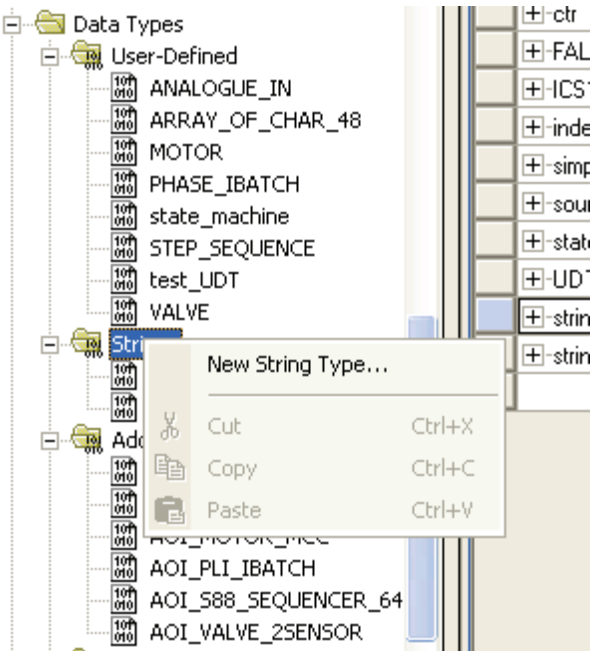
O cabeçalho de dados mostra como os grupos são definidos. O comprimento do grupo é inserido entre colchetes [] após o tipo de dados do grupo. O valor inicial do grupo é digitado na coluna “Initial Value”.

É possível criar um vetor de grupos, mas a cada um não pode ser dado um valor inicial. Uma definição alternativa para evitar esse problema é mostrada pela entrada “table” no cabeçalho de dados. “Table” é uma estrutura. O conteúdo dessa estrutura, não mostrado, são cinco instâncias de grupo[46], cada uma com um valor inicial.

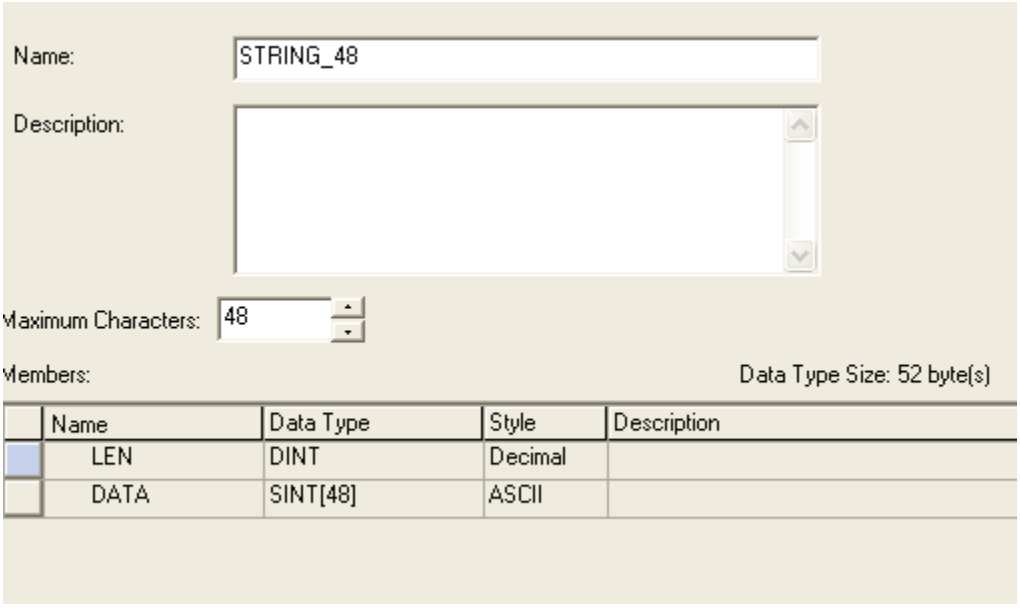
Definição de grupo no Logix

O trecho da tabela de configuração de tags abaixo mostra como os grupos são definidos no Logix.

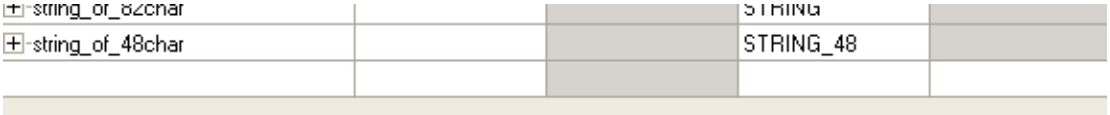
Se você desejar criar um grupo de um comprimento diferente dos 82 caracteres padrão, clique com o botão direito em “strings” em sua árvore de projeto (como mostrado abaixo).



Então, configure as propriedades conforme mostrado abaixo.



Feito isso, você pode definir instâncias do novo tipo.



Com instâncias do tipo STRING ou STRING_48, há um campo LEN atualizado automaticamente quando uma constante de grupo é inserida ou quando o grupo é manipulado por instruções ASCII ou STRING.

Variáveis temporárias STEP 7

Uma das categorias de variável no STEP 7 é a variável temporária. Elas podem ser criadas em qualquer bloco de organização, função ou bloco de funções.

As variáveis temporárias são usadas para armazenamento local temporário de valores intermediários e indicadores. Elas existem apenas enquanto seu bloco estiver sendo executado e seus valores são perdidos quando o bloco é extinto.

O Logix não tem variáveis temporárias. Todo armazenamento é estático, ou seja, os valores são mantidos entre as execuções de código.

Se você usa instruções add-on, você notará que as variáveis locais podem ser criadas para uma instrução add-on. Essas variáveis podem ser usadas da mesma maneira que as variáveis temporárias.

Funções

Se o programador em STEP 7 usar uma lista de declarações, ele pode ter que desenvolver rotinas de baixo nível demoradas de escrever e que requerem testes cuidadosos. As funções são importantes porque o desenvolvimento de tais rotinas precisa ser feito apenas uma vez e, depois de concluídas, tanto o originador da função quanto os outros programadores poderão fazer a mesma coisa em uma fração do tempo.

Esta seção descreve como as funções podem ser implementadas no Logix.

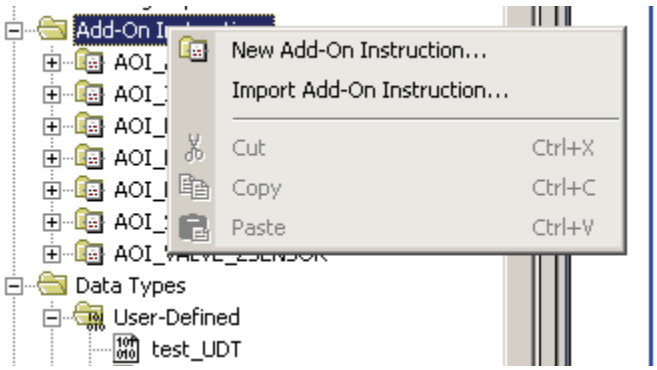
Funções como instruções add-on no Logix

As funções do STEP 7 e blocos de funções são semelhantes em sua estrutura às instruções add-on do Logix. As instruções add-on têm os mesmos tipos de parâmetros que os blocos de funções (Input, Output e InOut) e têm sua própria área de dados. Após codificada e testada, uma instrução add-on pode ser usada de qualquer lugar em um programa e é suficientemente autônoma para ser exportada a outros projetos ou colocada em uma biblioteca de códigos.

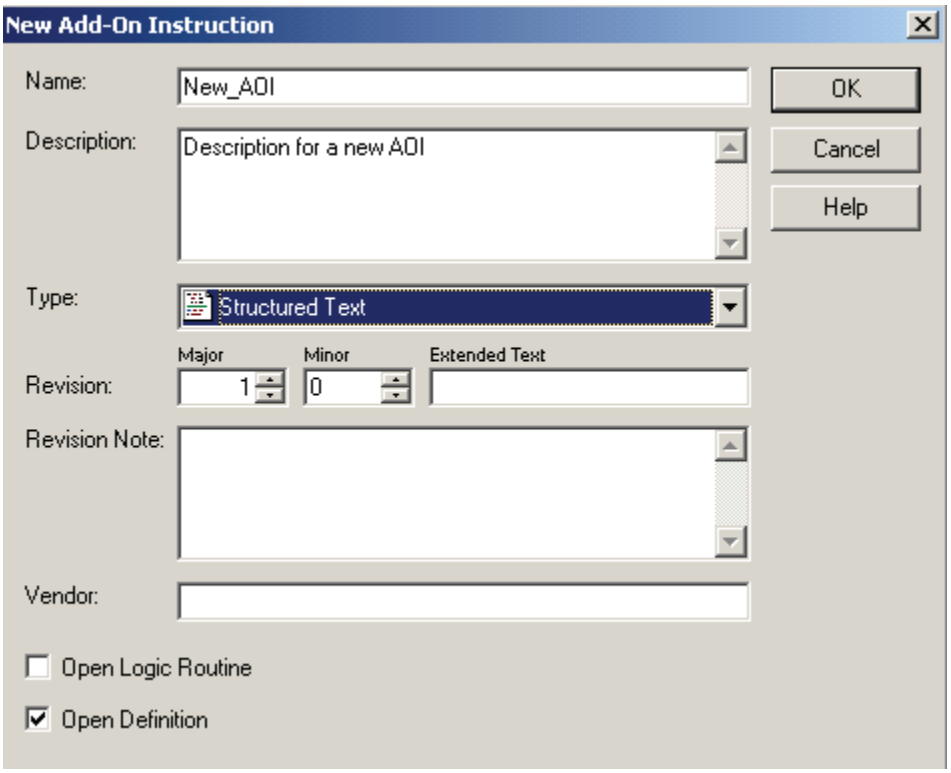
Exemplo - uma função de rampa

Este exemplo pega uma variável real e a acelera em rampa linearmente, partindo de seu valor inicial até um novo valor em uma taxa especificada.

Acesse a ramificação Add-On Instructions de sua árvore de projeto e clique com o botão direito em Add-On Instruction.

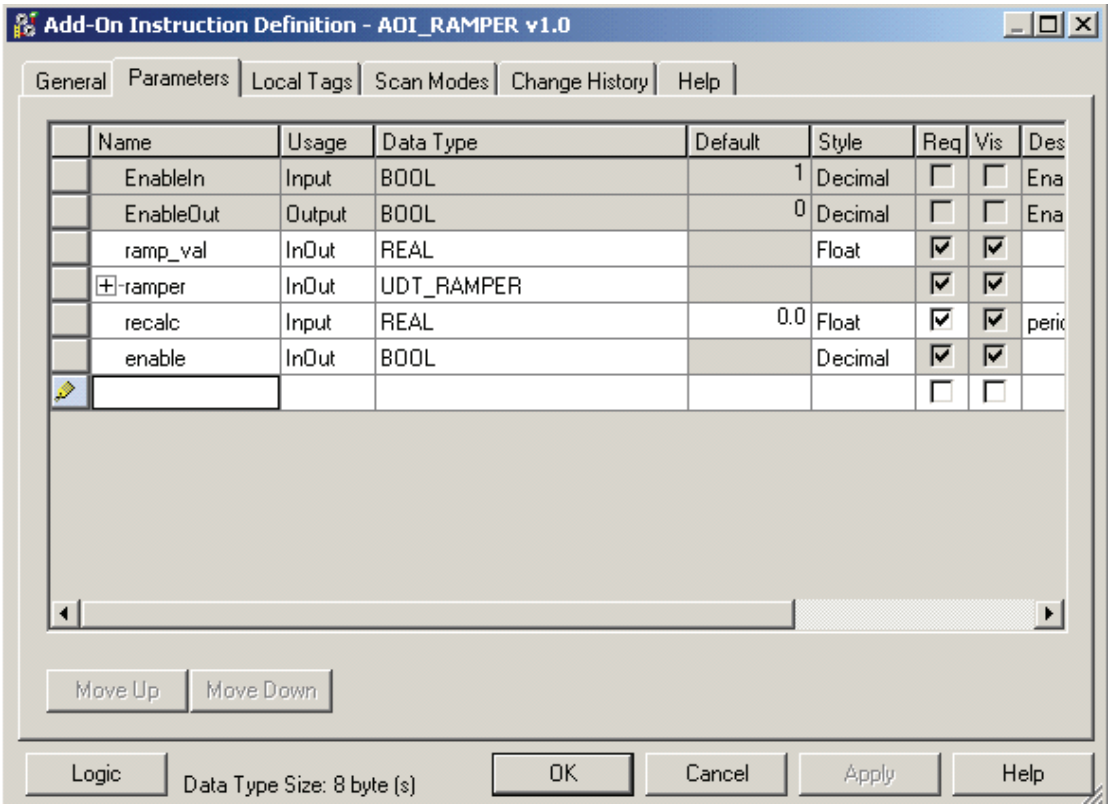


Este formulário será exibido.

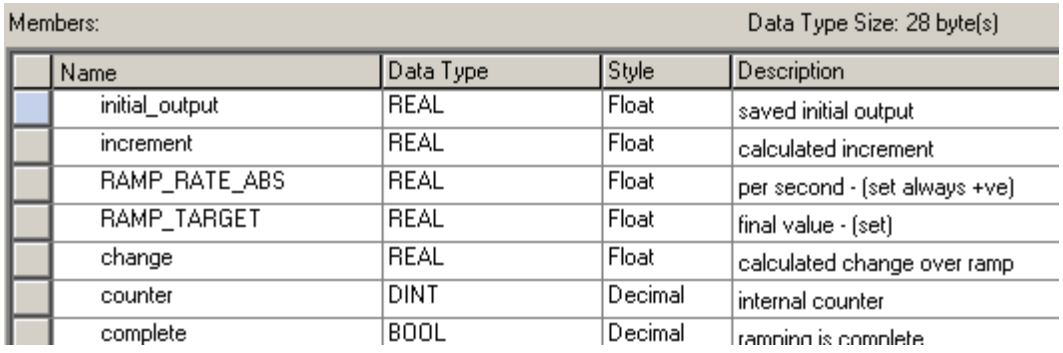


Digite o nome da instrução add-on e especifique a linguagem em que sua seção de código será escrita.

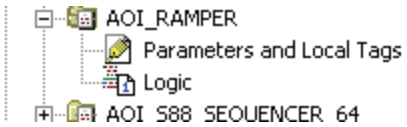
Escolha a guia Parameters.



Como no STEP 7, os parâmetros Input são os valores do programa para a instrução add-on, os parâmetros Output são valores da instrução add-on para o programa e os parâmetros InOut são para variáveis que serão modificadas pela instrução add-on. Se você tiver quaisquer estruturas de dados, escolha o tipo InOut, porque elas serão passadas por referência e isso é mais eficiente.



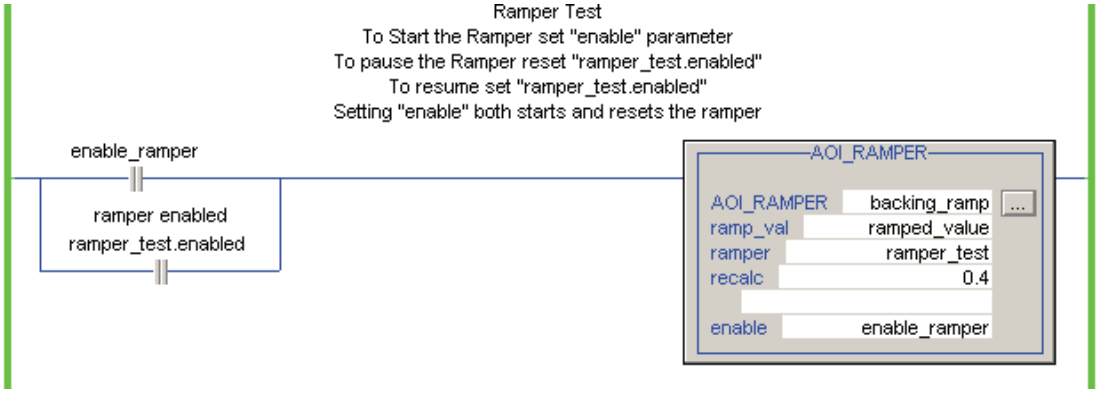
Na árvore de projeto para AOI_RAMPER, há uma seção lógica.



Abra-a para verificar o código dessa instrução add-on.

```
// Ramps a real variable from its current valuse to a new value at a
// specified rate.

// Parameters:
//      ramp_val    - variable to be ramped
//      ramper      - instance of UDT UDT_RAMPER
//      recalc      - code recalculation period (s)
//      enable      - start signal

// To use - set the target value in ramper.RAMP_TARGET_ABS
//        - set the ramp rate in ramper.RAMP_RATE_ABS
//        - to Start the Ramper set "enable" parameter
//        - to pause the Ramper reset "ramper.enabled"
//        - to resume set "ramper.enabled"
//        - setting "enable" both starts and resets the ramper

// on completion, the UDT field "complete" is set and the UDT field
// "enabled" is reset

// when enable is set, initialise
if (enable & (enable xor ramper._enable)) then
    ramper.initial_output := ramp_val;
    ramper.change := ramper.RAMP_TARGET - ramp_val;
    ramper.increment := ramper.change / abs(ramper.change)
                                * ramper.RAMP_RATE_ABS * recalc;
    ramper.counter := 0;
    ramper.complete := 0;
    enable := 0;
    ramper.enabled := 1;
end_if;

ramper._enable := enable;

// ramp calculations
if (ramper.enabled) then
    ramp_val := ramper.initial_output + (ramper.counter
                                            * ramper.increment);
    ramper.counter := ramper.counter + 1;
    if  (abs(ramper.counter * ramper.increment)
                                        > abs(ramper.change)) then
         ramp_val := ramper.RAMP_TARGET;
         ramper.complete := 1;
         ramper.enabled := 0;
    end_if;
end_if;
```

A instrução add-on pode ser chamada de qualquer rotina.



Observe que com instruções add-on você precisa criar um tag do tipo Add-On Instruction em uma área de dados visível à rotina. Isso é chamado de tag backing.

Antes de escrever uma instrução add-on, verifique a ajuda para instruções no software RSLogix 5000. Você pode descobrir uma instrução já existente que fará o trabalho. A seção a seguir ilustrará isso.

Cópia de bloco, COP e CPS

No STEP 7, é comum usar a função de sistema SFC20 “BLKMOV” para copiar um bloco de dados entre locais.

```
CALL  "BLKMOV"
 SRCBLK :="Data_EM1".stepMsgs.step5
 RET_VAL:=#intVar
 DSTBLK :="Data_EM1".actualStep
```

A instrução copia a string do quinto local em um vetor para um grupo de destino.

Freqüentemente queremos copiar o i-ésimo elemento de um vetor, sendo que “i” pode variar à medida que o programa é executado. “BLKMOV” não pode fazer isso.

O programador STEP 7 pode escrever uma função para atender a esse requisito.

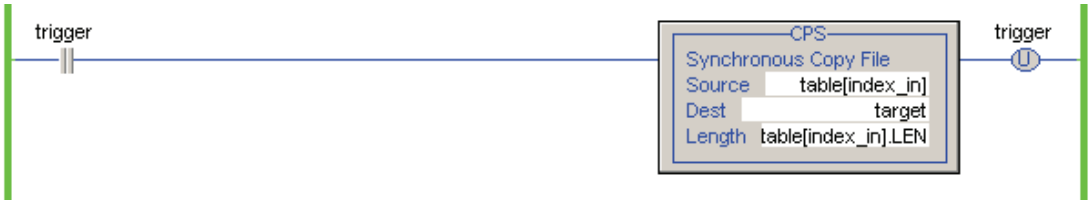
```
// copy step number descriptor to SCADA display area (EM faceplate)
      CALL  "INDEXED_COPY"
       indexSource :="Data_EMs".EM1.stepNumber
       sourceRef   :="Data_EM1".stepMsgs
       indexDest   :=1
       destRef     :="Data_EM1".actualStep
       recordLength:=8                  // 32 bytes
```

Nesse caso, a cópia está entre dois vetores e os índices são definidos por indexSource e indexDest.

No Logix, a instrução COP incorporada gravará todo o trabalho.

Como as especificações de origem e destino podem incluir índices de vetor variáveis, COP fará o trabalho. Esse é o equivalente de “INDEXED_COPY”.

A instrução CPS é a mesma que COP, mas com uma diferença.



A instrução não pode ser interrompida. Portanto, os dados de origem e destino permanecerão constantes ao longo da execução. Se você desejar mover dados que podem mudar, use CPS.

Os exemplos são:

- copiar dados de entrada para um buffer, de onde o programa operará sobre os dados.
- copiar tags consumidos para um buffer, de onde o programa operará sobre os dados.

Expressões matemáticas

Esta seção descreverá como o programador em S7 pode executar computações matemáticas no Logix. Um exemplo será usado – a expressão "v(cos(x)^2 + sin(x)^2)". O resultado dessa expressão é sempre exatamente 1, de modo que é fácil verificar se você está obtendo a resposta correta.

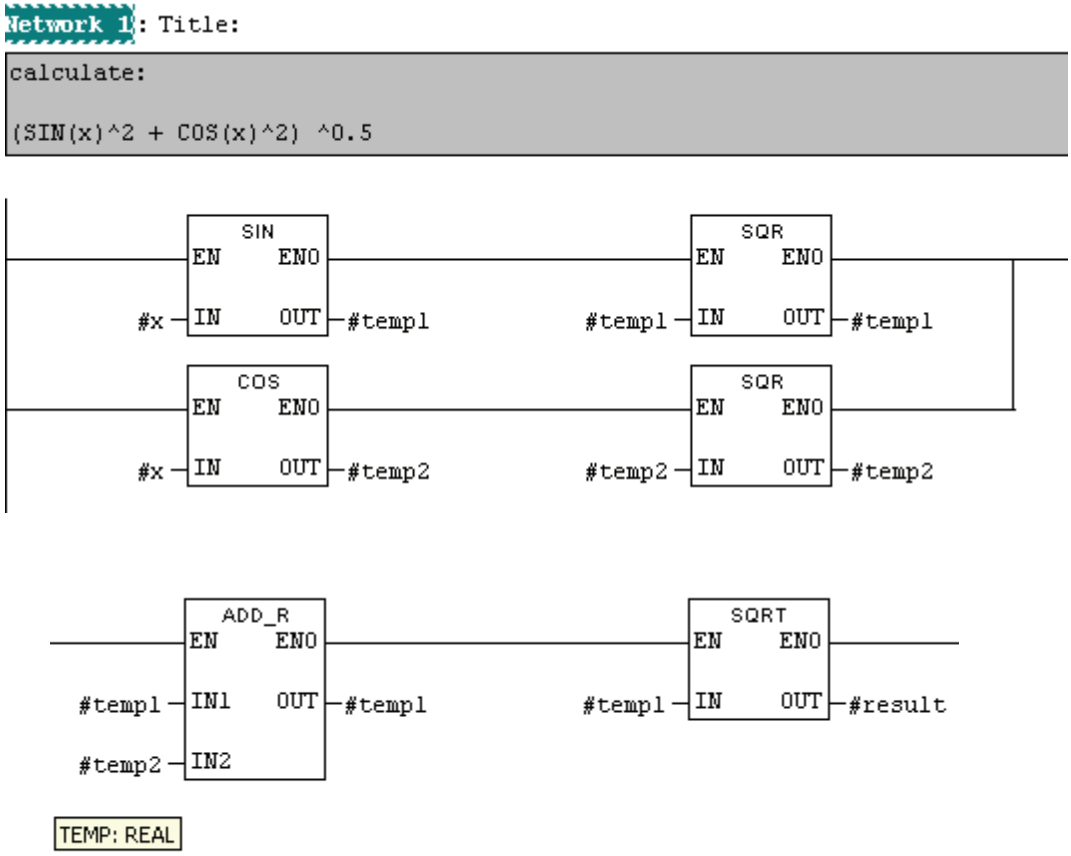
STEP 7 - STL

O código matemático no STL STEP 7 é eficiente, mas às vezes não muito claro para alguém que não esteja familiarizado com STL.

```
Network 1: Title:
calculate:

(SIN(x)^2 + COS(x)^2) ^0.5

      L     #x
      SIN
      SQR
      L     #x
      COS
      SQR
      +R
      SQRT
      T     #result
```

STEP 7 - LAD

A avaliação matemática em LAD segue um padrão convencional de combinação de funções.

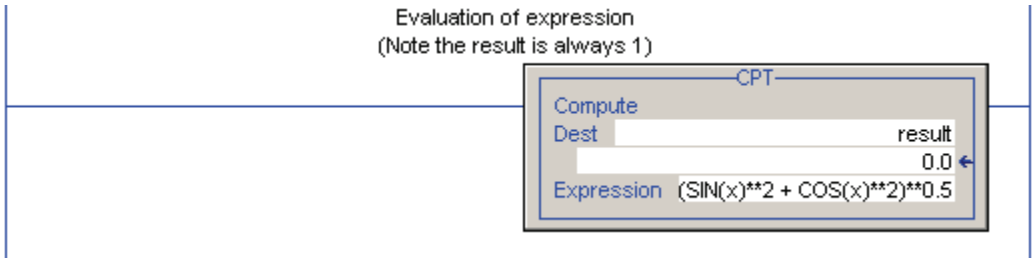


Logix - ST

A expressão é inserida da mesma maneira que com qualquer outra linguagem de alto nível.

```
// evaluation of mathematical expression in Structured Text

result := (SIN(x)**2 + COS(x)**2)**0.5;
```

Logix - LD



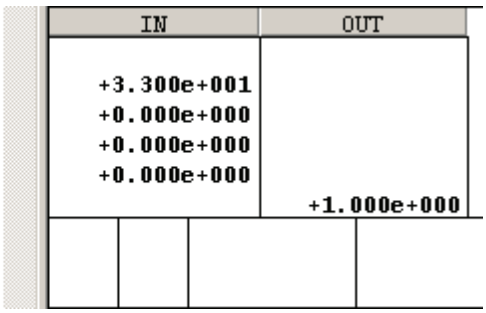
A instrução CPT permite que a expressão seja inserida de um modo de alto nível, que a maioria das pessoas entenderá mais facilmente que uma rede (linha) de instruções separadas.

STEP 7 - funções de usuário

Este bloco de funções foi escrito para fazer o mesmo que o CPT Logix.

```
CALL  "CPT" , "Data_CPT"                  FB119 / DB119
 str   :="test_formulae".test_string15    P#DB16.DBX476.0
 x     :=3.300000e+001
 a     :=0.000000e+000
 b     :=0.000000e+000
 c     :=0.000000e+000
 result:=#realVar
```

```
-- ((cos x)^2 + (sin x)^2) ^0.5 --
```



Ele lê e avalia um grupo de expressões armazenadas em um bloco de dados. Ele tem uma limitação comparado ao CPT Logix – a expressão é escrita em notação polonesa reversa, o que não atende a todo mundo.

Escrever um bloco de funções como esse cria problemas, sendo os principais o longo tempo e não ser para programadores iniciantes. Com o Logix, a instrução CPT está disponível para qualquer um usar assim que o software RSLogix 5000 seja instalado.

Verificação de tipo

Tanto no STEP 7 quanto no Logix, os parâmetros para funções, blocos de funções, instruções e instruções add-on são verificados estritamente quanto ao tipo por seus compiladores.

Há diferenças com expressões numéricas.

O Logix faz distinção entre valores numéricos e booleanos. O compilador rejeitará expressões que misturem de maneira não lógica valores numéricos e booleanos. Ao encontrar expressões de tipo numérico misturadas, ele fará conversões para produzir um resultado do tipo de variável de resultado declarado. Portanto, ele interpretará * como uma multiplicação de número inteiro se o resultado deve ser um número inteiro e como multiplicação real se o resultado deve ser um número real.

No STEP 7, o tipo de operações aritméticas deve ser especificado. Há, por exemplo, *I (multiplicar dois números inteiros com 16 bits) *D (multiplicar dois números inteiros de 32 bits) e *R (multiplicar dois números reais). Cabe ao programador assegurar que os dois números operandos de uma instrução *R sejam números reais. Se eles não forem, o compilador não reclamará, mas o resultado será sem sentido.

Conclusão

Os métodos Logix de programação de expressões matemáticas são mais claros e, separando o código matemático de outra lógica, torna o teste e a validação mais simples.

Outros tópicos relacionados à programação

Escopo de variáveis

Esta é uma área na qual o Logix difere consideravelmente do STEP 7.

Regras para o STEP 7

- As variáveis temporárias são invisíveis fora do bloco no qual elas foram declaradas.
- As variáveis estáticas globais são visíveis por todo o programa.
- As variáveis estáticas declaradas como dados de instância para um bloco de funções têm um status especial no bloco de funções, mas elas podem ser acessadas de outras partes do programa.

Regras para o Logix

A execução no Logix é dividida em tarefas. Cada tarefa pode ter diversos programas e cada programa pode ter diversas rotinas. Cada programa pode ter sua própria seção de tag.

- Os códigos de aceso de escopo do controlador são visíveis por todas as rotinas em todos os programas.
- Os tags de escopo do programa são visíveis apenas nas rotinas no programa no qual eles foram definidos. Isso significa que se uma rotina em um programa irá compartilhar dados com uma rotina em outro programa, ela deve usar dados de escopo do controlador.
- Os tags locais de instrução add-on são visíveis apenas pela lógica da instrução add-on.

OBs, tarefas e seqüenciamento

Os blocos de organização, tarefas e seqüenciamento são descritos no Capítulo 2.

Um exemplo maior - módulo de controle

Este exemplo montará alguns dos diferentes tópicos ilustrados nas seções anteriores. O termo “módulo de controle” (CM) provém do padrão de controle de batelada S88 influencial. O S88 encorajou o design de software de controlador a ser mais “orientado a objetos”. Esse módulo de controle é para uma válvula binária. A instrução add-on é adequada para esse tipo de programação.

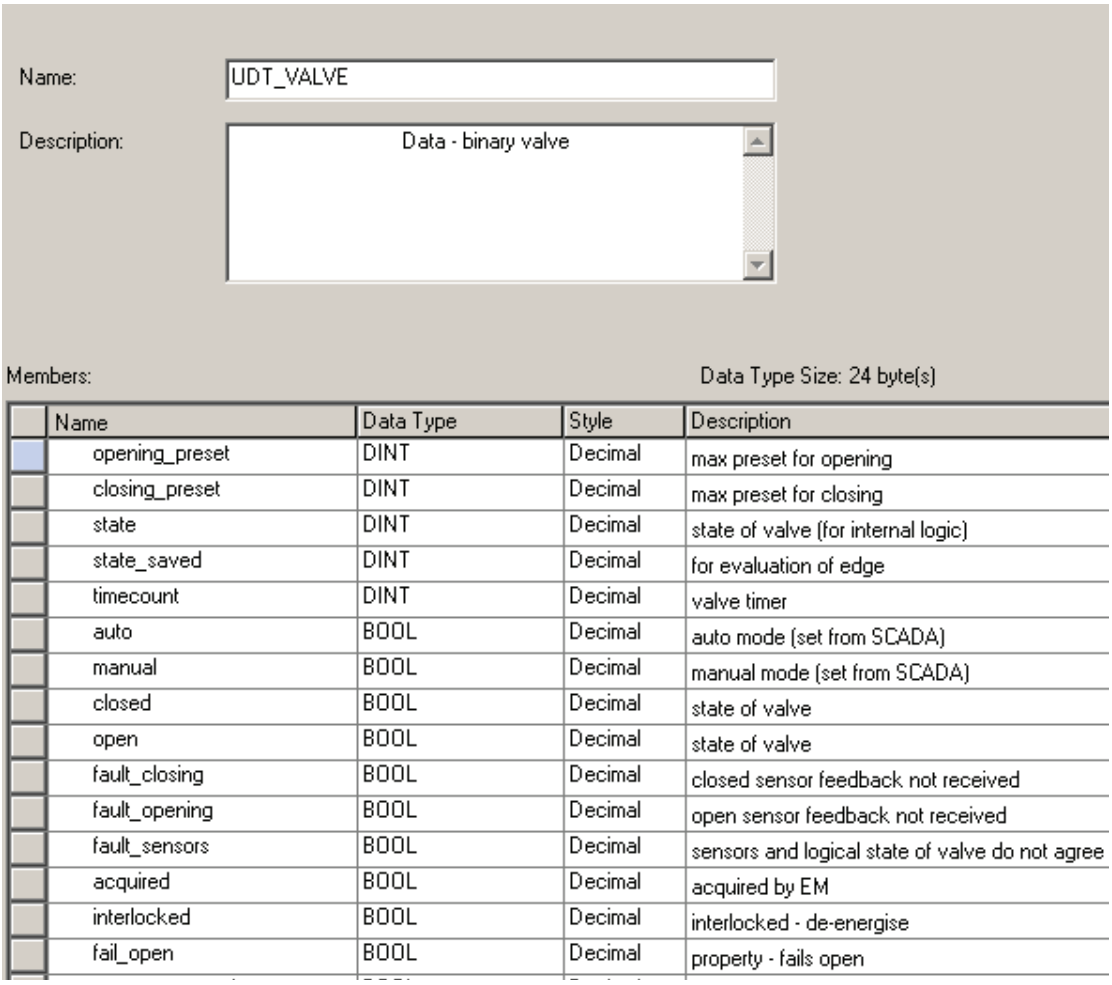
Componentes do CM

São:

- um UDT chamado UDT_VALVE.
- uma instrução add-on chamada AOI_VALVE_2SENSOR
- um novo programa em “task_02s” chamado “valves_callup”, que contém a seção e uma rotina de tags de programa.

Válvula de tipo de dados do usuário

O UDT é mostrado abaixo.



Name: UDT_VALVE

Description: Data - binary valve

Members: Data Type Size: 24 byte(s)

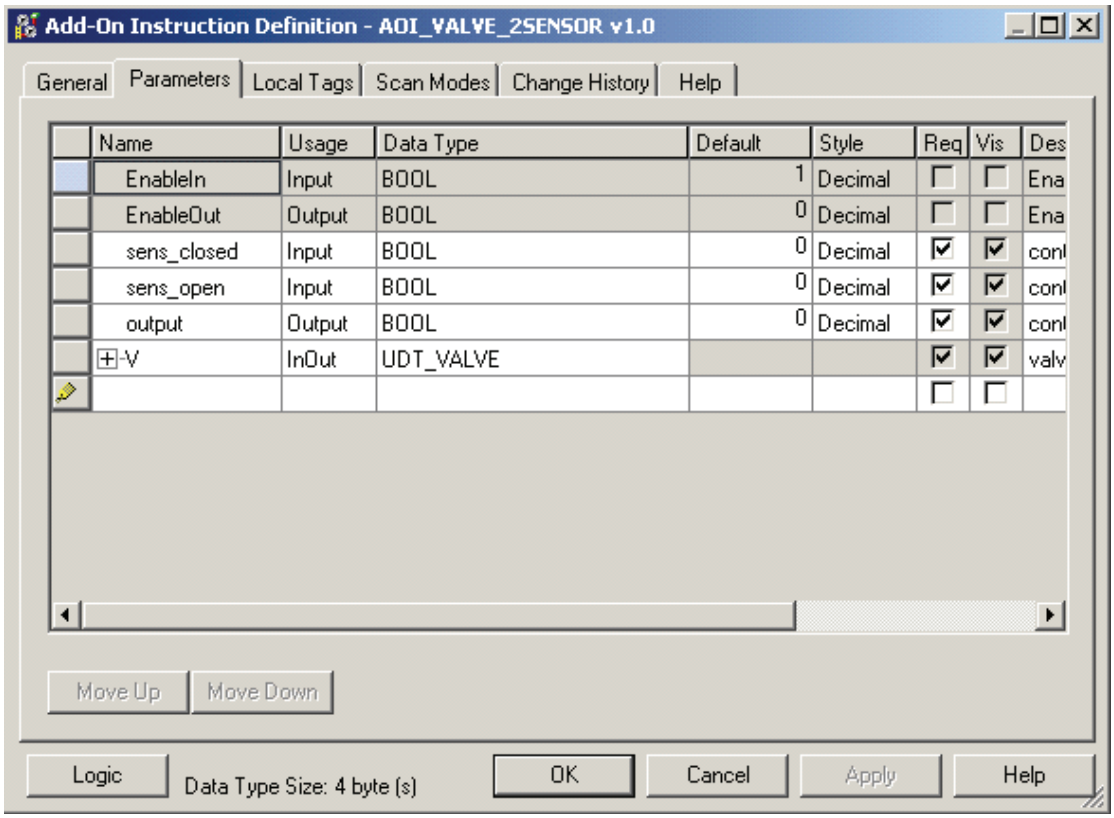
Name	Data Type	Style	Description
opening_preset	DINT	Decimal	max preset for opening
closing_preset	DINT	Decimal	max preset for closing
state	DINT	Decimal	state of valve (for internal logic)
state_saved	DINT	Decimal	for evaluation of edge
timecount	DINT	Decimal	valve timer
auto	BOOL	Decimal	auto mode (set from SCADA)
manual	BOOL	Decimal	manual mode (set from SCADA)
closed	BOOL	Decimal	state of valve
open	BOOL	Decimal	state of valve
fault_closing	BOOL	Decimal	closed sensor feedback not received
fault_opening	BOOL	Decimal	open sensor feedback not received
fault_sensors	BOOL	Decimal	sensors and logical state of valve do not agree
acquired	BOOL	Decimal	acquired by EM
interlocked	BOOL	Decimal	interlocked - de-energise
fail_open	BOOL	Decimal	property - fails open

Criar o UDT deve ser o primeiro passo – ele inclui todos os dados necessários para modelar a válvula.

A instrução add-on

Parâmetros da instrução add-on

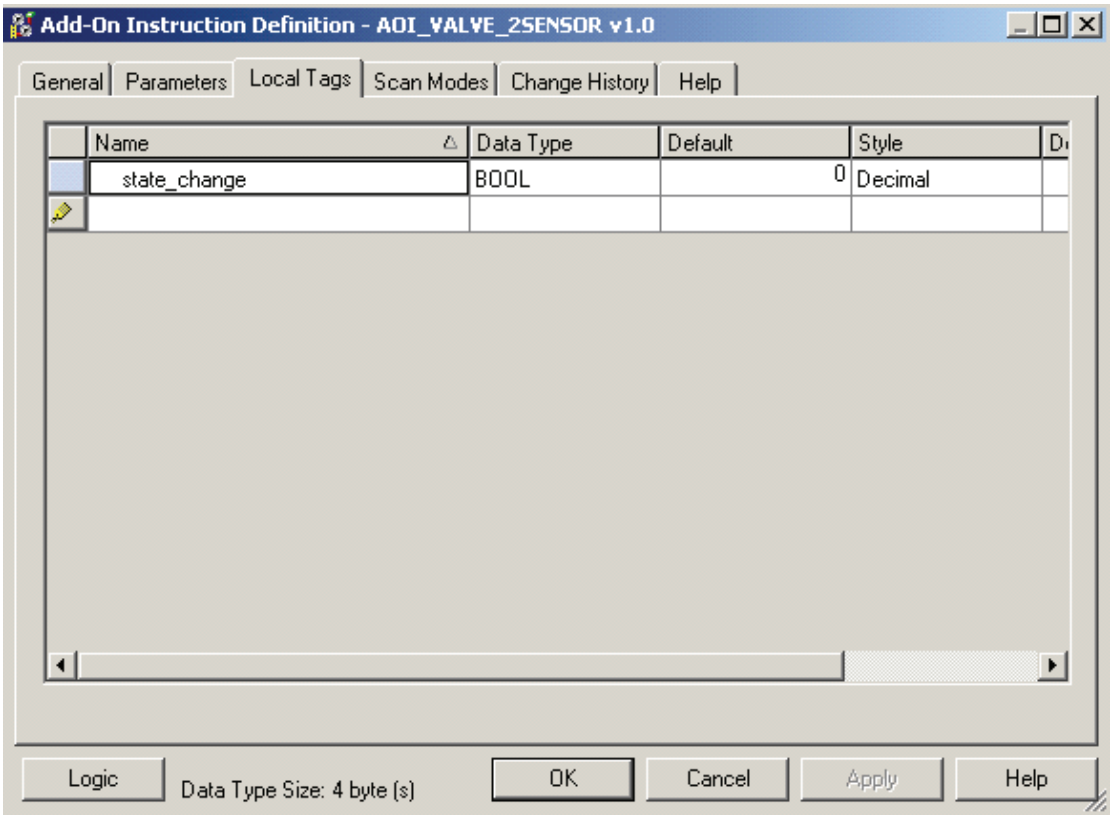
A captura de tela mostra a tela de configuração de parâmetros.



Os parâmetros adicionados são a E/S da válvula e um objeto do tipo "UDT_VALVE". "V" deve ser um parâmetro InOut.

Dados locais da instrução add-on

A captura de tela abaixo mostra a página de configuração dos dados locais da instrução add-on.



Lógica da instrução add-on

A captura de tela abaixo mostra a lógica dessa instrução add-on.

```
// Control Module Valve_2sensor
// ----------------------------
// Implements logic for a valve with an open and a closed sensor and one output

// See UDT Valve for data structure.

// Note the open/close command V.open_command must be set or reset externally
// and then left until the next activation is required. Do not continuously
// hold the flag set or reset.

// increment timer counter
V.timecount := V.timecount + 1;

// evaluate change of state (state machine)
state_change := V.state <> V.state_saved;
V.state_saved := V.state;

// set output
output := (V.fail_open xor V.open_command) and not
                                        (V.interlocked or V.faulted);

// valve is faulted
V.faulted := V.fault_opening or V.fault_closing or V.fault_sensors;

// action on fault or interlock

if V.faulted or V.interlocked then
    if V.fail_open then
        V.state := 3;
        V.open_command := 1;
    else
        V.state := 0;
        V.open_command := 0;
    end_if;
end_if;


// state machine:
// the state machine does not set outputs - it monitors inputs
// to set status and faults.
case V.state of
    // state 0 - valve is closed - wait for open command
    0:  V.closed := 1;
        V.open := 0;
        if (V.open_Command) then
            V.state := 1;
        // fault sensors
        else
            V.fault_sensors := (not sens_closed) or (sens_open);
        end_if;
    // state 1 -
    1:  V.state := 2;
    // state 2 - waiting for open sensor
    2:  if (sens_open & not sens_closed) then
            V.state := 3;
```

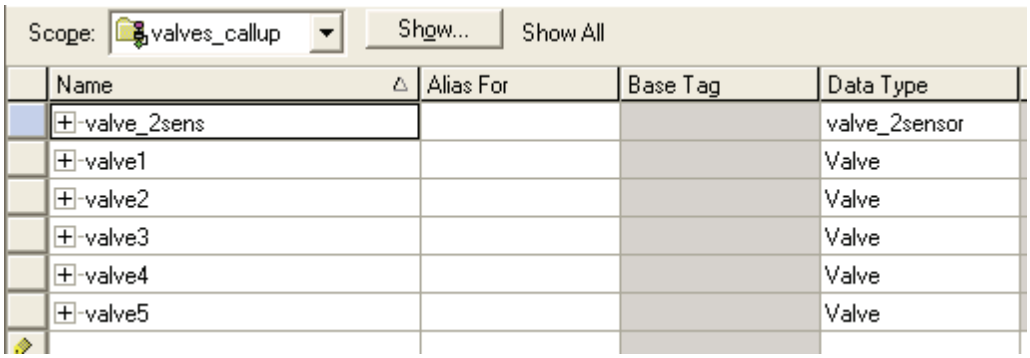
```
        // possible close command while waiting to open
        elsif not V.open_command then
            V.state := 0;
        // fault opening
        else
            V.fault_opening := (V.timecount >  V.opening_preset);
        end_if;
    // state 3 - open - wait for close command
    3:  V.closed := 0;
        V.open := 1;
        if (not V.open_command) then
            V.state := 4;
        // fault sensors
        else
            V.fault_sensors := (sens_closed) or (not sens_open);
        end_if;
    // state 4 -
    4:  V.state := 5;
    // state 5 - wait for closed sensor
    5:  if (sens_closed & not sens_open) then
            V.state := 0;
        // possible open command while waiting to close
        elsif V.open_command then
            V.state := 3;
        // fault closing
        else
            V.fault_closing := (V.timecount >  V.closing_preset);
        end_if;

          _
else;
end_case;
// end state machine

// reset timer if change of state
if (state_change) then V.timecount := 0;
end_if;

// external fault reset
if (V.clear_faults) then
    V.fault_opening := 0;
    V.fault_closing := 0;
    V.fault_sensors := 0;
    V.clear_faults := 0;
end_if;
```

Os tags referidos nessa lógica são todos parâmetros ou tags locais. Isso significa que a instrução add-on pode ser usada em qualquer programa (desde que UDT Valve esteja presente).

Chamada

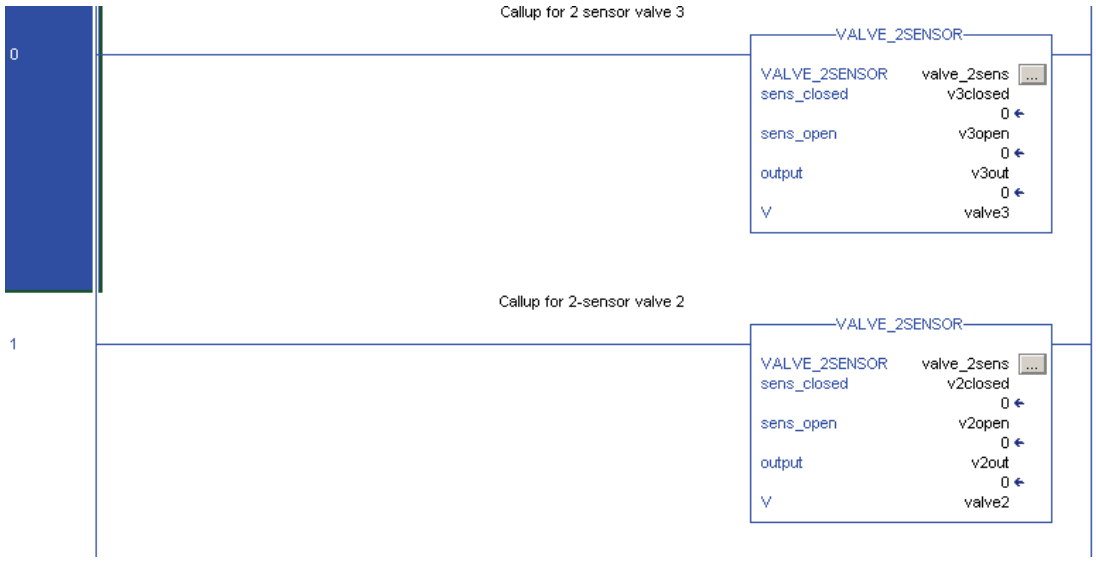
Tanto o código de chamada quanto as instâncias de UDT Valve estão localizadas no programa “valves_callup”, que é executado sob task_02s. A freqüência com que o código de chamada é executado depende da aplicação e do tamanho da válvula.

A captura de tela abaixo mostra as instâncias de dados.



Adicione uma instância do tipo Valve para cada válvula física. O primeiro tag é o “tag backing” necessário para a instrução add-on.

A captura de tela abaixo mostra o código de chamada.



Chame a instrução add-on uma vez para cada válvula. Os parâmetros reais são os tags de e/S reais para os sensores e solenóide da válvula e a instância de UDT "valve".

Os tags E/S aparecerão apenas na chamada da instrução add-on. Eles não serão usados em qualquer outro lugar do programa. Além de mais organizado do ponto de vista da estrutura do software, isso cancela qualquer risco de problemas provenientes da atualização assíncrona de E/S.

Lembre-se de que com os controladores Logix a E/S são varridas assincronamente.

Erros comuns ao converter para Logix

Introdução

O objetivo desta seção é apontar alguns dos erros de design e programação que os usuários do S7 cometem freqüentemente ao converter aplicações para o Logix. Esses erros foram identificados pela examinação de programas Logix convertidos a partir do STEP 7.

Tópico	Página
Não selecionar hardware apropriado	129
Subestimar o seqüenciamento de tarefas	130
Executar a tradução em lugar da conversão	130
Não usar as linguagens Logix mais apropriadas	130
Implementação de tipos de dados incorretos – DINT versus INT	131
Código de usuário simulando instruções existentes	132
Uso incorreto de COP, MOV e CPS	133
Uso incorreto de CPT	133
Não manusear grupos de maneira ideal	133
Uso extensivo de saltos	133
Não usar tags com alias	133

Os erros de programação estão nestas duas categorias:

- Programação que leva à redução de eficiência do controlador.
- Programação que leva a um sistema de controle difícil de compreender, manter e desenvolver.

Na maioria dos casos, a codificação para eficiência também aprimorará a facilidade de leitura e modularidade de seu programa. Reciprocamente, aprimorar a estrutura do programa também o torna mais eficiente.

Não selecionar hardware apropriado

Este capítulo concentra-se principalmente em software. Lembre-se, entretanto, de que a seleção correta de hardware é um requisito para a operação satisfatória. É possível que o número de controladores e racks não seja o mesmo que para um sistema S7 equivalente.

Leia o Capítulo 1 e Apêndice A para obter mais informações sobre hardware. Mais informações podem ser encontradas nos Apêndices A e B.

Subestimar o seqüenciamento de tarefas

Na área de seqüenciamento e interrupções, não há muita diferença na capacidade dos dois sistemas. Entretanto, no mundo do Logix, o seqüenciamento é encorajado mais ativamente.

É muito comum que os programadores em STEP 7 negligenciem o seqüenciamento ao trabalharem com os controladores Logix. Consulte o Capítulo 2 para uma explicação mais detalhada sobre seqüenciamento no Logix.

Executar a tradução em lugar da conversão

É um erro comum traduzir um programa em STEP 7 linha por linha para o Logix.

Em vez disso, um processo mais completo é necessário, o qual é descrito como conversão. Isso abrange a escolha de linguagens, seqüenciamento e escolha de rotinas de código.

Ao converter seus programas em STEP 7 em vez de traduzi-los, você estará fazendo um uso melhor da capacidade de seu sistema Logix.

Não usar as linguagens Logix mais apropriadas

Os programadores freqüentemente negligenciam as linguagens Logix que não a lógica ladder.

Leia o Capítulo 2 para uma explicação de como escolher uma linguagem Logix e o Capítulo 4 para obter exemplos de código STEP 7 convertido para o Logix.

Implementação de tipos de dados incorretos – DINT versus INT

É comum aconselhar o uso de DINT em vez de INT.

O exemplo abaixo mostra uma adição de dois DINTs *versus* a adição de dois INTs.

Adicionar DINTs

```
// add two DINTs and assign to a third DINT

for index := 0 to 999 do
    result_DINT := operandA_DINT + operandB_DINT;
end_for;
```

Adicionar INTs

```
// add two INTs and assign to a third INT

for index := 0 to 999 do
    result_INT := operandA_INT + operandB_INT;
end_for;
```

Resultados de temporização

A tabela mostra tempos relativos (número menor é mais rápido). Os números aqui são apenas para comparação com outros números na tabela. Eles não devem ser comparados com entradas em outras tabelas.

Método	Tempos relativos
Adicionar DINTs com ST para loop	53
Adicionar INTs com ST para loop	100

Para comparação, o mesmo teste foi feito com um controlador S7. Nesse caso, os resultados foram idênticos para DINTs e INTs.

A lição é usar DINT para que todos os números inteiros funcionem no Logix. Use apenas INT ou SINT se estiver fazendo interface com um sistema externo que requeira o uso de INTs ou SINTs.

Código de usuário simulando instruções existentes

Os programadores freqüentemente escrevem código de usuário quando uma instrução existente faria o trabalho. Como exemplo, compare a cópia de um vetor com o código de usuário na instrução COP.

Código de usuário

```
for index := 0 to 99 do
    target_DINT[index] := source_DINT[index];
end_for;
```

Instrução COP

```
cop(source_DINT[0], target_DINT[0], 100);
```

Abaixo estão os tempos relativos de dois métodos. Novamente, os números são apenas para comparação com outros números na tabela. Eles não devem ser comparados com entradas em outras tabelas.

Método	Tempos relativos
Cópia de vetor de DINTs com texto estruturado	100
Cópia de vetor de DINTs com COP	18

Para executar operações como copiar vetores, as funções de biblioteca do STEP 7 escritas na lista de declarações são usadas. Se a função de biblioteca não fizer o necessário, uma nova pode ser escrita. As funções escritas podem ser quase tão eficientes quando aquelas que o STEP 7 fornece.

Entretanto, no Logix, é possível que um programador escreva uma função de cópia que seja tão eficiente quando a COP incorporada. A lição para programadores em S7 é verificar a ajuda sobre instruções no software RSLogix 5000 cuidadosamente.

Uso incorreto de COP, MOV e CPS

MOV copia um simples valor (imediato ou tag) para um tipo de tag simples – DINT, INT, SINT ou REAL. COP pode fazer o mesmo que MOV (a origem não pode ser um valor imediato), mas seu uso mais importante é copiar tipos de dados complexos.

Seria um erro de programação menor usar o COP para copiar tipos de dados simples.

Um erro visto mais freqüentemente é usar vários MOVs para copiar uma estrutura de dados quando um COP poderia ser usado.

Se seus dados de origem puderem mudar durante a cópia devido a atualização de E/S assíncronas, use o CPS. Essa instrução não pode ser interrompida de modo de dados de origem permanecerão constantes durante a cópia.

Uso incorreto de CPT

No Logix, a instrução CPT pode ser usada para avaliar expressões. A expressão é inserida em um dos campos da instrução. Isso é muito conveniente.

Entretanto, a CPT deve apenas ser usada se mais de uma instrução aritmética for necessária para avaliar a expressão. Se uma única instrução for suficiente, ela será mais rápida que a CPT.

Você pode ler mais a respeito da CPT no Capítulo 4.

Não manusear grupos de maneira ideal

Se você quiser definir um novo tipo de grupo, por exemplo, com um número diferente de caracteres que os 82 padrão, pode ser um erro criar um novo 'tipo de dados do usuário'. Em vez disso, crie um novo tipo de dados de grupo. A vantagem de fazer isso dessa maneira é que o campo 'LEN' será atualizado automaticamente à medida que o comprimento do grupo mudar.

Uso extensivo de saltos

No Logix, os saltos podem ocorrer apenas na lógica ladder. É recomendado que a instrução JMP seja usada moderadamente. Os saltos na lógica ladder freqüentemente tornam o programa mais difícil de ler.

Não usar tags com alias

Lembre-se de criar tags com alias par aos tags de E/S que o software RSLogix 5000 cria para você. Eles tornarão o programa mais fácil de ler. Consulte o Capítulo 2.

Observações:

Glossário S7 para Logix

Introdução

Este capítulo fornece um glossário dos termos S7 e seus equivalentes no Logix.

Terminologia de hardware

Termo S7	Definição	Termo Logix	Definição
Processador de comunicações	Módulo de comunicações	Ponte	
Controlador	O controlador	Controlador	
CPU	Unidade central de processamento	CPU ou controlador	
CPU Fail safe	CPU 315F-2 DP implementa a versão PROFISAFE de DP	GuardLogix	L61S, L62S, L63S
Ethernet industrial	Versão Siemens da Ethernet	EtherNet/IP ControlNet	Ambos têm a mesma (ou melhor) funcionalidade que a Ethernet industrial
MPI	Interface multiponto – um barramento serial	Serial	Protocolos DF1 ou DH485
Controlador programável		Controlador ou PAC	
PROFIBUS DP	Barramento de campo usado comumente	EtherNet/IP ControlNet DeviceNet	
PROFIBUS PA	Variação de Profibus especializado em automação de processo	Como DP Profibus	
PROFINET	Profibus pela Ethernet	EtherNet/IP	
PROFISAFE	Versão livre de falhas do DP PROFIBUS	GuardLogix	
S7-200	Controladores de baixa tecnologia	MicroLogix	
S7-300	Controladores de nível intermediário	CompactLogix	
S7-400	Controladores de alta tecnologia	ControlLogix	
SIMATIC	Nome comercial pra produtos de automação da Siemens	Logix	

Terminologia de software

Termo S7	Definição	Termo Logix mais aproximado	Definição
Acumulador	Usado em STL	Não aplicável	Nas linguagens Logix, não há necessidade de acessar instruções de baixo nível da CPU
AR1, AR2	Registradores de indicador	Não aplicável	Nas linguagens Logix, não há necessidade de acessar instruções de baixo nível da CPU
Vetor	Sintaxe VETOR[0…7] DE REAL	Vetor	Sintaxe REAL[8] A indexação começa sempre em 0
Memória de bit	Endereços M...	Não aplicável	Usa tags
Transferência de blocos	Copia bloco de dados. SFC20 BLK_MOV	COP	Instrução (usa MOV para uma variável simples)
BOOL		BOOL	
Byte	Palavra de 8 bits	SINT	Seu uso é desencorajado (é mais lento que DINT) exceto quando necessário (por exemplo, caracteres de grupo)
CFC	Linguagem de controle de processo opcional	FBD	Linguagem de bloco de funções padrão
CHAR	Byte como caractere	SINT	
Cycle_Execution	OB1 – executado continuamente	tarefa contínua	Executado continuamente
Bloco de dados	Unidade de memória de dados estáticos	Banco de dados de tags de escopo do controlador ou banco de dados de tags de escopo do programa	Global visível no programa ao qual o banco de dados está vinculado
DINT	Número inteiro duplo	DINT	Número inteiro duplo
DWORD	Palavra de 32 bits	DINT	
FBD	diagrama de blocos de funções	FBD	diagrama de blocos de funções
Função	Unidade de programa com memória temporária, mas sem memória estática	Rotina AOI (instrução add-on)	Ambas **podem** corresponder a uma função
Bloco de funções	Unidade de programa com memória temporária e estática	Rotina AOI (instrução add-on) Programa	Todas **podem** corresponder a um bloco de funções
GRAPH	Linguagem gráfica opcional	controle seqüencial de funções	Linguagem gráfica padrão
Configuração de hardware	Configuração de hardware – componente do STEP 7	Configuração de E/S	Ramificação do organizador do controlador
INT	Número inteiro	INT	Seu uso é desencorajado (é mais lento que DINT)
Interrupt_Execution	OB executado periodicamente OB	tarefa periódica	Tarefa executada periodicamente
LAD	lógica ladder	LD	lógica ladder
Biblioteca	Funções do sistema	GSV, SSV	Instruções – Obter valor do sistema Definir valor do sistema

Termo S7	Definição	Termo Logix mais aproximado	Definição
NetPro	Configurador de rede	Não aplicável	Parte da ramificação de configuração de E/S do organizador do controlador
Bloco de organização	Unidade de programa chamada pelo sistema operacional	Tarefa	Unidade de programa chamada pelo sistema operacional
Indicador	Indicador de dados usado no STL	Não aplicável	Usa vetores
REAL	Número de ponto flutuante de 32 bits	REAL	Número de ponto flutuante de 32 bits
SCL	Linguagem de alto nível opcional	texto estruturado	Linguagem padrão
Gerenciador Simatic	Componente do STEP 7	Organizador do controlador	Componente do RSLogix 5000
STEP 7	Software de monitoração e desenvolvimento para S7	RSLogix 5000	Software de monitoração e desenvolvimento para Logix
STL	Lista de declarações	Não aplicável	Usar texto estruturado ou lógica ladder ou controle seqüencial de funções
GRUPO	Seqüência de caracteres Comprimento padrão 254	GRUPO	Seqüência de SINTs. Comprimento padrão 82. O objeto de grupo contém também seu comprimento como propriedade.LEN
ESTRUTURA	Conjunto sem tipo de dados	Não aplicável	No Logix, uma estrutura é uma instância de tipo (UDT)
Símbolo	Nome para endereço de memória de dados	Tag	O tag define a estrutura da variável *e* reserva memória
Memória temporária	Memória criada na pilha no momento da execução	Não aplicável	Usa tags
PALAVRA	Palavra de 16 bits	INT	
UDT	Tipo de dados do usuário	UDT	Tipo de dados do usuário

Observações:

Peças de S7 300 e S7 400 e equivalentes RA

Introdução

Este apêndice relaciona os produtos Siemens e seus equivalentes Rockwell Automation.

Tópico	Página
CPUs S7 300 compactas	140
CPUs S7 300 padrão	140
CPUs com tecnologia S7 300	141
CPUs S7 300 Fail-Safe	141
Módulos de entrada digital S7 300	142
Módulos de saída digital S7 300	142
Módulos de saída a relé S7 300	143
Módulos digitais combinados S7 300	143
Módulos de entrada analógica S7 300	144
Módulos de saída analógica S7 300	145
Módulos combinados analógicos S7 300	145
Módulos de saída analógica S7 300	146
Controladores redundantes e livres de falhas	146
Módulos de entrada digital	147
Módulos de saída digital	147
Módulos de entrada analógica	147
Módulos de saída analógica	148

CPUs S7 300 compactas

Nº catálogo Siemens	Referência rápida Siemens	Memória		Portas de comuni-cação		Tamanho máximo MMC		E/S incorporada			Solução RA
			MPI	DP	Série		DI	DO	AI	AO	
6ES7 312-5BE0x-xxxx	S7-312C	32K	S	N	N	4 MB	10	6			1769-L31 + E/S compacta ML1500
6ES7 313-5BF0x-xxxx	S7-313C	64K Sim Não Não	S	N	N	8 MB	24	16	4	2	1769-L31 + E/S compacta ML1500
6ES7 313-6BF0x-xxxx	S7-313C- PtP	64K	S	N	RS422/485	8 MB	16	16			1769-L31 + E/S compacta ML1500
6ES7 313-6CF0x-xxxx	S7-313C- DP	64K	S	S	N	8 MB	16	16			1769-L31 + E/S compacta ML1500
6ES7 314-6BG0x-xxxx	S7-314C- PtP	96K	S	N	RS422/485	8 MB	24	16	4	2	1769-L31 + E/S compacta ML1500
6ES7 314-6CG0x-xxxx	S7-314C- DP	96K Sim Sim Não 8 MB	S	S	N	8 MB	24	16	4	2	1769-L31 + E/S compacta ML1500

CPUs S7 300 padrão

Siemens Nº catálogo	Referência rápida Siemens	Memória		Portas de comunicação		Tamanho de memória de carga máxima (RAM)	Solução RA
			MPI	DP	PN		
6ES7 312-1AE1x-xxxx	S7-312	32K	S	N	N	4 MB	1769-L31
6ES7 314-1AG1x-xxxx	S7-314	96K	S	N	N	8 MB	1769-L31
6ES7 315-2AG1x-xxxx	S7-315-2 DP	128K	S	S	N	8 MB	1769-L3xE ou 1769-L3xC
6ES7 315-2EH1x-xxxx	S7-315-2 PN/DP	256K	S	S	S	8 MB	1769-L3xE ou 1769-L3xC

Siemens Nº catálogo	Referência rápida Siemens	Memória		Portas de comunicação		Tamanho de memória de carga máxima (RAM)	Solução RA
			MPI	DP	PN		
6ES7 317-2AJ1x-xxxx	S7-317-2 DP	512K	S	S	N	8 MB	1769-L3xE ou 1769-L3xC
6ES7 317-2EK1x-xxxx	S7-317-2 PN/DP	1 MB	S	S	S	8 MB	1769-L3xE ou 1769-L3xC
6ES7 319-3EL0x-xxxx	S7-319-3 PN/DP	1,4 MB	S	S	S	8 MB	1769-L3xE ou 1769-L3xC

CPUs com tecnologia S7 300

Siemens Nº Catálogo	Referência rápida Siemens	Memória		Portas de comunicação		Tamanho de memória de carga máxima (RAM)	Solução RA
			MPI	DP	PN		
6ES7 315-6TG1x-xxxx	S7-315T-2 DP	128K	S	S	S	4 ou 8 MB	1768-L43
6ES7 317-6TJ1x-xxxx	S7-317T-2 DP	512K	S	S	S	4 ou 8 MB	1768-L43

CPUs S7 300 Fail-Safe

Siemens Nº Catálogo	Referência rápida Siemens	Memória		Portas de comunicação		Tamanho de memória de carga máxima (RAM)	Solução RA ControlLogix
			MPI	DP	PN		
6ES7 315-6FF1x-xxxx	S7-315F-2 DP	192K	S	S	N	8 MB	GuardLogix ou SmartGuard 600
6ES7 315-2FH1x-xxxx	S7-315F-2 PN/DP	256K	S	S	S	8 MB	GuardLogix ou SmartGuard 600
6ES7 317-6FF0x-xxxx	S7-317F-2 DP	1 MB	S	S	N	8 MB	GuardLogix ou SmartGuard 600
6ES7 317-2FK1x-xxxx	S7-317F-2 PN/DP	1 MB	S	S	S	8 MB	GuardLogix ou SmartGuard 600

Módulos de entrada digital S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Faixa	Solução RA	Comentários
6ES7 321-1BH0x-xxxx	20 pinos	16	24 Vcc	1769-IQ16 1769-IQ16F	
6ES7 321-1BH5x-xxxx	20 pinos	16	24 Vcc	1769-IQ16 1769-IQ16F	
6ES7 321-1BL0x-xxxx	40 pinos	32	24 Vcc	1769-IQ32 1769-IQ32T	
6ES7 321-1CH0x-xxxx	40 pinos	16	24 . . . 48 V	não aplicável	
6ES7 321-1CH2x-xxxx	20 pinos	16	48 . . . 125 Vcc	não aplicável	
6ES7 321-1BH1x-xxxx	20 pinos	16	24 Vcc	1769-IQ16 1769-IQ16F	
6ES7 321-7BH0x-xxxx	20 pinos	16	24 Vcc	1769-IQ16 1769-IQ16F	
6ES7 321-1FH0x-xxxx	20 pinos	16	120 . . . 230 Vca	1769-IA16	1769-IA16 admite apenas 120 Vca
6ES7 321-1FF0x-xxxx	20 pinos	8	120 ... 230 Vca	1769-IM12	1769-IM12 admite apenas 230 Vca
6ES7 321-1FF1x-xxxx	40 pinos	8	120 ... 230 Vca	1769-IA8I	1769-IA8I admite apenas 120 Vca
6ES7 321-1EL0x-xxxx	40 pinos	32	120 Vca	não aplicável	
não aplicável		16	5 Vcc TTL	1769-IG16	

Módulos de saída digital S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Faixa	Corrente de saída	Solução RA	Comentários
6ES7 332-1FH0x-xxxx	20 pinos	16	120/230 Vca	0,5 A	1769-OA16	
6ES7 332-1FF0x-xxxx	20 pinos	8	120/230 Vca	2 A	1769-OA8	S7-300 possui fusível por grupo
6ES7 332-5FF0x-xxxx	40 pinos	8	120/230 Vca	2 A	1769-OA8	S7-300 vem em grupos de 1
6ES7 322-1BH0x-xxxx	20 pinos	16	24 Vcc	0,5 A	1769-OB16 1769-OB16P	
6ES7 322-1BH1x-xxxx	20 pinos	16	24 Vcc	0,5 A	não aplicável	Alta velocidade

6ES7 322-1BL0x-xxxx	40 pinos	32	24 Vcc	0,5 A	1769-OB32 1769-OB32T	
6ES7 322-1BF0x-xxxx	20 pinos	8	24 Vcc	2 A	1769-OB8	
6ES7 322-8BF0x-xxxx	20 pinos	8	24 Vcc	0,5 A	1769-OB8	
6ES7 332-1FL0x-xxxx	2 x 20 pinos	32	120 Vca	1 A	não aplicável	
6ES7 332-5GH0x-xxxx	40 pinos	16	24/48 V	0,5 A	não aplicável	
6ES7 332-1CF0x-xxxx	20 pinos	8	48 … 125 Vcc		não aplicável	
não aplicável		16	5 Vcc TTL		1769-OG16	
não aplicável		16	24 Vcc		1769-OV16	
não aplicável		32	24 Vcc		1769-OV32T	
não aplicável		16	24 Vcc		1769-OB16P	

Módulos de saída a relé S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Corrente de saída	Solução RA	Comentários
6ES7 322-1HH0x-xxxx	20 pinos	16	2 A	1769-OW16	
6ES7 322-1HF0x-xxxx	20 pinos	8	5 A	1769-OW8	
6ES7 322-1HF1x-xxxx	40 pinos	8	5 A	1769-OW8I	
6ES7 322-5HF0x-xxxx	40 pinos	8	8 A	1769-OW8I	O módulo S7-300 vem com filtro RC e proteção contra sobretensão

Módulos digitais combinados S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Entradas da faixa	Corrente de saída	Solução RA	Comentários
6ES7 323-1BH0x-xxxx	20 pinos	8 / 8	24 Vcc	24 Vcc / 0,5 A	1769-IQ6XOW4	E/S compacta tem menos E/S e as saídas possuem relé
6ES7 323-1BL0x-xxxx	40 pinos	16 / 16	24 Vcc	24 Vcc / 0,5 A	não aplicável	
6ES7 327-1BH0x-xxxx	20 pinos	8 / 8	24 Vcc	24 Vcc / 0,5 A	não aplicável	8 entradas; 8 entradas ou saídas (configurável)

Módulos de entrada analógica S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Resolução (bits)	Tipo	Solução de E/S compacta	Comentários
6ES7 331-1KF0x-xxxx	40	8	13	Tensão, corrente, resistência Temperatura	1769sc-IF8U 1769-IF8U	
6ES7 331-7KF0x-xxxx	20	8	9 / 12 / 14	Tensão, corrente, resistência Temperatura	1769sc-IF8U 1769-IF8U	
6ES7 331-7 kB0x-xxxx	20	2	9 / 12 / 14	Tensão, corrente, resistência Temperatura	1769sc-IF8U 1769-IF4	
6ES7 331-7NF0x-xxxx	40	8	16	Tensão Tensão	1769-IF8	
6ES7 331-7NF1x-xxxx	40	8	16	Tensão Tensão	1769-IF8	Inclui interrupção de hardware ao final do ciclo vs 6ES7 331-7NF0x-xxxx
6ES7 331-7HF0x-xxxx	20	8	14	Tensão Tensão	1769-IF8	
6ES7 331-7PF0x-xxxx	40	8		RTD Resistência	1769-IR6	
6ES7 331-7PF1x-xxxx	40	8		Termopar	1769-IT6	
não aplicável					1769-IF4I	

Módulos de saída analógica S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Resolução (bits)	Tipo	Solução RA	Comentários
6ES7 332-5HD0x-xxxx	40	4	12	Tensão Corrente	1769-OF4VI 1769-OF4CI	
6ES7 332-7ND0x-xxxx	20	4	16	Tensão Corrente	1769-OF4VI 1769-OF4CI	
6ES7 332-5HB0x-xxxx	20	2	12	Tensão Corrente	1769-OF2	
6ES7 332-5HF0x-xxxx	20	8	12	Tensão Corrente	1769-OF8V 1769-OF8C	

Módulos combinados analógicos S7 300

Número de catálogo Siemens	Conector frontal	Pontos	Resolução (bits)	Tipo	Solução RA	Comentários
6ES7 334-0KE0x-xxxx	20	4 / 2	12	Tensão Corrente Pt 100		Apenas saídas tensão
6ES7 334-0CE0x-xxxx	20	4 / 2	8	Tensão e corrente (entradas e saídas)	1769-IF4XOF2	

Controladores S7 400 padrão

Siemens Nº Catálogo	Referência rápida Siemens	Tamanho da memória de trabalho		Portas de comunicação		Tamanho de memória de carga máxima (RAM)	Solução RA ControlLogix
			MPI	DP	PN		
6ES7 412-1XF04-0AB0	CPU 412-1	144 kB	S	S	N	64 MB	1756-L61
6ES7 412-2GX04-0AB0	CPU 412-2	256 kB	S	S	N	64 MB	1756-L61
6ES7 414-2GX04-0AB0	CPU 414-2	512 kB	S	S	N	64 MB	1756-L62
6ES7 414-3XJ04-0AB0	CPU 414-3	1,4 MB	S	S	N	64 MB	1756-L63
6ES7 414-3EM05-0AB0	CPU 414-3 PN/DP	2,8 MB	S	S	S	64 MB	1756-L63
6ES7 416-3XK04-0AB0	CPU 416-2	2,8 MB	S	S	N	64 MB	1756-L63
6ES7 416-3XL04-0AB0	CPU 416-3	5,6MB	S	S	N	64 MB	1756-L64
6ES7 416-3ER05-0AB0	CPU 416-3 PN/DP	11,2 MB	S	S	S	64 MB	1756-L64
6ES7 417-4XL04-0AB0	CPU 417-4	20 MB	S	S	N	64 MB	1756-L64

Controladores redundantes e livres de falhas

Número de catálogo Siemens	Referência rápida Siemens	Tamanho da memória de trabalho		Portas de comunicação			Tamanho de memória de carga máxima (RAM)	Solução RA ControlLogix
			MPI	DP	PN	Portas de sincronização		
6ES7 414-4HJ04-0AB0	CPU 414-4H	1,4 MB	S	S	N	S	64 MB	1756-L63
6ES7 417-4HL04-0AB0	CPU 417-4H	20 MB	S	S	N	S	64 MB	1756-L64
6ES7 416-2FK04-0AB0	CPU-416F-2	2,6 MB	S	S	N	N	64 MB	1756-L61S

Módulos de entrada digital

Número de catálogo Siemens	Conector frontal	Pontos	Faixa	Solução RA	Comentários
6ES7 421-7BH01-0AB0 (interrupção/ diagnóstico)	48 pinos	16	24 Vcc	1756-IB16D	
6ES7 421-1BL01-0AA0	48 pinos	32	24 Vcc	1756-IB32	
6ES7 421-1EL00-0AA0	48 pinos	32	120 Vca/cc	1756-IA32	
6ES7 421-1FH20-0AA0	48 pinos	16	230 Vca/cc	1756-IM161	
6ES7 421-7DH00 0AB0 (interrupção/ diagnóstico)	48 pinos	32	24-60 Vca/cc		

Módulos de saída digital

Número de catálogo Siemens	Conector frontal	Pontos	Faixa	Corrente	Solução RA	Comentários
6ES7 422-1FH00-0AA0	48 pinos	16	230 Vca	2 A	1756-0A16	
6ES7 422-1HH00-0AA0	48 pinos	16	60 Vcc 230 Vca (relê)	5 A	1756-OW16I	
6ES7 422 1BH11-0AA0	48 pinos	16	24 Vcc	2 A	1756-0B16E	
6ES7 422-1BL00-0AA0	48 pinos	32	24 Vcc	0,5 A	1756-0B32	
6ES7 422-7BL00-0AB0 (diagnóstico)	48 pinos	32	24 Vcc	0,5 A	1756-0B16D 1756-0B32	

Módulos de entrada analógica

Número de catálogo Siemens	Conector frontal	Canais	Resolução (bits)	Tipo	Solução RA	Comentários
6ES7 431-0HH0-0AB0	48 pinos	16	13	Tensão Corrente	1756-IF16	16 bits
6ES7 431-1KF00-0AB0	48 pinos	8	13	Tensão Corrente Impedância	1756-IF8	16 bits 4 entradas diferenciais
6ES7 431-1KF10-0AB0	48 pinos	8	14-16	Tensão Corrente Termopar Termorresistor Impedância	1756-IR6I 1756-IT6I	6 RTD 6 Termopar Ambos 16 bit

6ES7 431-1FK20-0AB0	48 pinos	8	14	Tensão Corrente Impedância	1756-IF16	16 bits
6ES7 431-7QH00-0AB0 (interrupção)	48 pinos	16	16	Tensão Corrente Termopar Termorresistor Impedância	1756-IR6I 1756-IT6I	6 RTD 6 Termopar
6ES7 431-7KF00-0AB0	48 pinos	8	16	Tensão Corrente Termopar	1756-IT6I	6 canais
6ES7 431-7KF01-0AB0	48 pinos	8	16	Termorresistor	1756-IR6I	5 canais

Módulos de saída analógica

Número de catálogo Siemens	Conector frontal	Canais	Resolução (bits)	Tipo	Solução RA	Comentários
6ES7 432-1HF00-0AB0	48 pinos	8	13	Tensão Corrente	1756-OF8	15 bits

Tabela de referência cruzada HMI Siemens

Use este apêndice para comparar os painéis Rockwell Automation com tipos de painéis específicos Siemens.

Tópico	Página
Micropainéis SIMATIC e equivalentes Rockwell Automation	149
Painéis SIMATIC - série 7x e equivalentes Rockwell Automation	151
Painéis SIMATIC - série 17x e equivalentes Rockwell Automation	152
Painéis SIMATIC - série 27x e equivalentes Rockwell Automation	154
Multipainéis SIMATIC - série 27x e equivalentes Rockwell Automation	156
Multipainéis SIMATIC - série 37x e equivalentes Rockwell Automation	158

Micropainéis SIMATIC e equivalentes Rockwell Automation

Micropainéis SIMATIC					Solução Rockwell Automation		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6640-0BA11-0AX0	SIMATIC OP 73MICRO	Visor monocromático de 3 pol. STN, 160x48 pixels, teclado, 24 Vcc apenas	128 KB	Compatível com 1xRS485, S7-200, sem porta de impressora	2711P-K4M5D	PanelView Plus 400 escala de cinza, teclado	Visor STN de 3,8 pol., escala de cinza de 32 níveis, 320 x 240 pixels, comunicação RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0AA15-2AX0	SIMATIC TP070 Finalizado em abril de 2007	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas	128 KB	Compatível com 1xRS485, S7-200, sem porta de impressora	2711P-T6M5D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320 x 240 pixels, RS-232 comunicação, toque, 24 Vcc, recursos de impressão USB
6AV6640-0CA01-0AX0	SIMATIC TP 170MICRO Finalizado em abril de 2007	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas, funcionalidade de aplicativo limitada	256KB	Compatível com 1xRS485, S7-200, sem porta de impressora	2711P-T6M5D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320x240 pixels, RS-232 comunicação, toque, 24 Vcc, recursos de impressão USB

Micropainéis SIMATIC					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6640-0CA11-0AX0	SIMATIC TP 177MICRO	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas	256KB	Compatível com 1xRS485, S7-200, sem porta de impressora	2711P-T6M5D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320x240 pixels, RS-232 comunicação, toque, 24 Vcc, recursos de impressão USB
6AV6610-0AA01-1CA8	Software WINCC FLEXIBLE MICRO	Software de configuração e programação para micropainéis Simatic apenas	não aplicável	não aplicável	9701-VWSTMENE	Software RSView Studio Machine Edition	Software de configuração RSView Studio for Machine Edition para desenvolvimento e teste de aplicativos HMI no nível da máquina

Painéis SIMATIC - série 7*x* e equivalentes Rockwell Automation

Painéis SIMATIC - série 7*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6641-0AA11-0AX0	SIMATIC OP73	Visor STN monocromático de 3 pol., 160x48 pixels, teclado, 24 Vcc apenas	256KB	1x RS485, S7-200, S7-compatível com 300/400, sem porta de impressora	2711P-K4M5D	PanelView Plus 400 escala de cinza, teclado	Visor STN de 3,8 pol., escala de cinza de 32 níveis, 320x240 pixels, RS-232 comunicação, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6641-0BA11-0AX0	SIMATIC OP77A	Visor STN monocromático de 4,5 pol., 160x64 pixels, teclado. 24 Vcc apenas	256KB	1xRS422, 1xRS485, S7-200, S7-300/400, sem porta de impressora	2711P-K4M5D	PanelView Plus 400 escala de cinza, teclado	Visor STN de 3,8 pol., escala de cinza de 32 níveis, 320x240 pixels, RS-232 comunicação, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6641-0CA01-0AX0	SIMATIC OP77B	Visor STN monocromático de 4,5 pol., 160x64 pixels, teclado, 24 Vcc apenas	1 MB	1xRS232, 1xRS422, 1xRS485, USB, S7-200, S7-300/400, porta de impressora disponível	2711P-K4M5D	PanelView Plus 400 escala de cinza, teclado	Visor STN de 3,8 pol., escala de cinza de 32 níveis, RS-232 comunicação, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6621-0AA01-0AA0	Software WINCC FLEXIBLE COMPACT	Software de programação e configuração para Simatic OP77, OP/TP170 e micropainéis	não aplicável	não aplicável	9701-VWSTMENE	Software RSView Studio for Machine Edition	Software de configuração RSView Studio Machine Edition para desenvolvimento e teste aplicativos HMI no nível de máquina

Painéis SIMATIC - série 17*x* e equivalentes Rockwell Automation

Painéis SIMATIC - série 17*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6545-0BA15-2AX0	SIMATIC TP170A modo Blue Finalizado em abril de 2007	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas	320 KB	1xRS232, 1xRS422, 1xRS485, S5, S7-200, S7-300/400, e controladores de terceiros, sem porta de impressora	2711P-T6M20D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320x240 pixels, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0BB15-2AX0	SIMATIC TP170B modo Blue Finalizado em abril de 2007	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas	768 KB	2xRS232, 1xRS422, 1xRS485, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T6M20D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320x240 pixels, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0BC15-2AX0	SIMATIC TP170B em cores Finalizado em abril de 2007	Visor STN de 5,7 pol. (256 cores), 320x240 pixels, toque. 24 Vcc apenas	768 KB	2xRS232, 1xRS422, 1xRS485, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T6C20D	PanelView Plus 600 em cores, toque	Visor TFT colorido de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6542-0BB15-2AX0	SIMATIC OP170B modo Blue Finalizado em abril de 2007	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, teclado e toque, apenas 24 Vcc	768 KB	2xRS232, 1xRS422, 1xRS485, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-B6M20D	PanelView Plus 600, escala de cinza, toque e teclado	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320x240 pixels, EtherNet/IP, comunicação RS-323, toque e teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6642-0DC01-1AX0	SIMATIC OP177B modo Blue	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, teclado e toque, apenas 24 Vcc	2 MB	1xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-B6M20D	PanelView Plus 600, escala de cinza, toque e teclado	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320x240 pixels, EtherNet/IP, comunicação RS-323, toque e teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6642-0AA11-0AX0	SIMATIC TP177A modo Blue	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas	512 KB	Compatível com 1xRS422, 1xRS485, S7-200, S7-300/400, sem porta de impressora	2711P-T6M20D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320 x 240, pixels, EtherNet/IP, comunicação RS-323, toque, 24 Vcc, 64 MB Flash, recursos de impressão USB

Painéis SIMATIC - série 17*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6642-0BA01-1AX0	SIMATIC TP177B em cores	Visor colorido STN de 5,7 pol. (256 cores), 320x240 pixels, toque. 24 Vcc apenas	2 MB	1xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T6C20D	PanelView Plus 600 em cores, toque	Visor colorido TFT de 5,5 pol.; 320 x 240 pixels, profundidade de cor de 18 bits, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB Flash, recursos de impressão USB
6AV6642-0BC01-1AX0	SIMATIC TP177B modo Blue	Visor STN de 5,7 pol., modo Blue (4 níveis), 320x240 pixels, toque, 24 Vcc apenas	2 MB	1xRS422, 1xRS485, USB, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T6M20D	PanelView Plus 600 escala de cinza, toque	Visor STN de 5,5 pol., escala de cinza de 32 níveis, 320 x 240, pixels, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB Flash, recursos de impressão USB
6AV6642-8BA10-0AA0	SIMATIC TP177B em cores, aço inoxidável	Visor colorido STN de 5,7 pol. (256 cores), 320x240 pixels, toque, apenas 24 Vcc, moldura de aço inoxidáel	2 MB	1xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T6C20D	PanelView Plus 600 em cores, toque	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6642-0DA01-1AX0	SIMATIC OP177B em cores	Visor colorido STN de 5,7 pol. (256 cores), 320x240 pixels, teclado e toque, 24 Vcc apenas	2 MB	1xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-B6C20D	PanelView Plus 600 em cores, toque,e teclado	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, comunicação RS-232, toque e teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6621-0AA01-0AA0	Software WINCC FLEXIBLE COMPACT	Software de configuração e programação para Simatic OP77, OP/TP170 e micropainéis	não aplicável	não aplicável	9701-VWSTMENE	Software RSView Studio Machine Edition	Software de configuração RSView Studio Machine Edition para desenvolvimento e teste aplicativos HMI no nível de máquina

Painéis SIMATIC - série 27x e equivalentes Rockwell Automation

Painéis SIMATIC - série 27*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6545-0CA10-0AX0	SIMATIC TP270 6 pol. em cores Finalizado em outubro de 2006	Visor colorido STN de 5,7 pol., (256 cores), 320x240 pixels, toque, 24 Vcc apenas	2 MB	2xRS232, 1xRS422, 1xRS485, USB, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressão disponível.	2711P-T6C20D	PanelView Plus 600 em cores, toque	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, Comunicação RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0CC10-0AX0	SIMATIC TP270 10 pol. em cores Finalizado em outubro de 2006	Visor colorido STN de 10,4 pol., (256 cores), 640x480 pixels, toque, 24 Vcc apenas	2 MB	2xRS232, 1xRS422, 1xRS485, USB, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T10C4D1	PanelView Plus 1000 em cores, toque	Visor TFT de 10,4 pol., 640x480 pixels, 18 bits em cores, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6542-0CA10-0AX0	SIMATIC OP270 6 pol. em cores Finalizado em outubro de 2006	Visor colorido STN de 5,7 pol., (256 cores), 320x240 pixels, teclado, 24 Vcc apenas	2 MB	2xRS232, 1xRS422, 1xRS485, USB, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-K6C20D	PanelView Plus 600 em cores	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, Comunicação RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6542-0CC10-0AX0	SIMATIC OP270 10 pol. em cores Finalizado em outubro de 2006	Visor colorido STN de 10,4 pol. (256 cores), 640x480 pixels, teclado, 24 Vcc apenas	2 MB	2xRS232, 1xRS422, 1xRS485, USB, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-K10C4D1	PanelView Plus 1000 em cores, teclado	Visor TFT de 10,4 pol., 640x480 pixels, profundidade de cor de 18 bits, EtherNet/IP e RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB

Painéis SIMATIC - série 27*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6643-0AA01-1AX0	SIMATIC TP 277 6 pol. em cores	Visor colorido STN de 5,7 pol., (256 cores), 320x240 pixels, toque, 24 Vcc apenas	4 MB	1xRS422, 1xRS485, USB, Ethernet: S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T6C20D	PanelView Plus 600 em cores, toque	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, comunicação RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6643-0BA01-1AX0	SIMATIC OP 277 6 pol., em cores	Visor colorido STN de 5,7 pol., (256 cores), 320x240 pixels, teclado, 24 Vcc apenas	4 MB	1xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-K6C20D	PanelView Plus 600 em cores	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, comunicação RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6622-0BA01-0AA0	Software WINCC FLEXIBLE STANDARD	Software de configuração e programação para Simatic OP/TP/ MP270, MP370, OP77, OP/TP170 e micropainéis	não aplicável	não aplicável	9701-VWSTMENE	Software RSView Studio Machine Edition	Software de configuração RSView Studio Machine Edition para desenvolvimento e teste aplicativos HMI no nível de máquina

Multipainéis SIMATIC - série 27*x* e equivalentes Rockwell Automation

Multipainéis SIMATIC - série 27*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6542-0AG10-0AX0	SIMATIC MP270B teclado, 10 pol. Finalizado em outubro de 2006	Visor colorido TFT de 10,4 pol. em cores (64 k), 640x480 pixels, teclado, 24 Vcc apenas	5 MB	2xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-K10C4D1	PanelView Plus 1000 em cores, teclado	Visor TFT de 10,4 pol., 640x480 pixels, profundidade de cor de 18 bits, EtherNet/IP e RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0AG10-0AX0	SIMATIC MP270B toque, 10 pol. Finalizado em outubro de 2006	Visor colorido TFT de 10,4 pol. (64 k), 640x480 pixels, toque, 24 Vcc apenas	5 MB	2xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T10C4D1	PanelView Plus 1000 em cores, toque	Visor TFT de 10,4 pol., 640x480 pixels, 18 bits em cores, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0AH10-0AX0	SIMATIC MP270B toque, 6 pol. Finalizado em outubro de 2006	Visor colorido TFT de 5,7 pol. (64 k), 320x240 pixels, toque, 24 Vcc apenas	5 MB	2xRS422, 1xRS485, USB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-K6C20D	PanelView Plus 600 em cores	Visor colorido TFT de 5,5 pol., 320x240 pixels, profundidade de cor de 18 bits, EtherNet/IP, Comunicação RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6643-0CB01-1AX0	SIMATIC MP 277 toque, 8 pol.	Visor colorido TFT de 7,5 pol. (64 k), 640x480 pixels, toque, 24 Vcc apenas	6 MB	1xRS422, 1xRS485, 2xUSB, Ethernet, S5, S7-200, S7-300/400 e controladores de terceiros, porta de impressora disponível	2711P-T7C4D1	PanelView Plus 700 em cores, toque	Visor TFT de 6,5 pol., 640x480 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6643-0CD01-1AX0	SIMATIC MP 277 toque, 10 pol.	Visor colorido TFT de 10,4 pol. (64 k), 640x480 pixels, toque, 24 Vcc apenas	6 MB	1xRS422, 1xRS485, 2xUSB, Ethernet: S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível	2711P-T10C4D1	PanelView Plus 1000 em cores, toque	Visor TFT de 10,4 pol., 640x480 pixels, 18 bits em cores, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB

Multipainéis SIMATIC - série 27*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
–	SIMATIC MP 277 toque, 10 pol., aço inoxidável	Visor colorido TFT de 10,4 pol. (64 k), 640x480 pixels, toque, 24 Vcc apenas, moldura de aço inoxidável, IP66	6 MB	1xRS422, 1xRS485, 2xUSB, Ethernet, S5, S7-200, S7-300/400 e controladores de terceiros, porta de impressora disponível	2711P-T10C4D1	PanelView Plus 1000 em cores, toque	Visor TFT de 10,4 pol., 640x480 pixels, 18 bits em cores, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6643-0DB01-1AX0	SIMATIC MP 277 teclado, 8 pol.	Visor colorido TFT de 7,5 pol. em cores (64 k), 640x480 pixels, teclado, 24 Vcc apenas	6 MB	1xRS422, 1xRS485, 2xUSB, Ethernet, S5, S7-200, S7-300/400 e controladores de terceiros, porta de impressora disponível	2711P-K7C4D1	PanelView Plus 700 em cores, teclado	Visor TFT de 6,5 pol., 640x480 pixels, profundidade de cor de 18 bits, EtherNet/IP e RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6643-0DD01-1AX0	SIMATIC MP 277, teclado, 10 pol.	Visor colorido TFT de 10,5 pol. em cores (64 k), 640x480 pixels, teclado, 24 Vcc apenas	6 MB	1xRS422, 1xRS485, 2xUSB, Ethernet, S5, S7-200, S7-300/400 e controladores de terceiros, porta de impressora disponível	2711P-K10C4D1	PanelView Plus 1000 em cores, teclado	1Visor TFT de 10,4 pol., 640x480 pixels, profundidade de cor de 18 bits, EtherNet/IP e RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6622-0BA01-0AA0	Software WINCC FLEXIBLE STANDARD	Software de configuração e programação para Simatic OP/TP/ MP270, MP370, OP77, OP/TP170 e micropainéis	não aplicável	não aplicável	9701-VWSTMENE	Software RSView Studio Machine Edition	Software de configuração RSView Studio Machine Edition para desenvolvimento e teste aplicativos HMI no nível de máquina

Multipainéis SIMATIC - série 37*x* e equivalentes Rockwell Automation

Multipainéis SIMATIC - série 37*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6542-0DA10-0AX0	SIMATIC MP370, teclado, 12 pol.	Visor colorido TFT de 12,1 pol. (256 cores), 800x600 pixels, teclado, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-K12C4D1	PanelView Plus 1250 em cores, teclado	Visor TFT de 12,1 pol., 800x600 pixels, cores de 18 bits, EtherNet/IP e RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0DA10-0AX0	SIMATIC MP370, toque, 12 pol.	Visor colorido TFT de 12,1 pol. (256 cores), 800x600 pixels, toque, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-T12C4D1	PanelView Plus 1250, em cores, toque	Visor TFT de 12,1 pol., 800x600 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-0DB10-0AX0	SIMATIC MP370 toque, 15 pol.	Visor colorido TFT de 15,1 pol. (256 cores), 1024x768 pixels, toque, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-T15C4D1	PanelView Plus 1500, em cores, toque	Visor TFT de 15 pol., 1024x768 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6545-8DB10-0AA0	SIMATIC MP370, toque, 15 pol., aço inoxidável	Visor colorido TFT de 15,1 pol. (256 cores), 1024x768 pixels, toque, 24 Vcc apenas, moldura de aço inoxidável, IP66	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-T15C4D1	PanelView Plus 1500, em cores, toque	Visor TFT de 15 pol., 1024x768 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6 644-0AA01-2AX0	SIMATIC MP377, toque 12,1 pol.	Visor TFT de 12,1 pol., 65.536 cores, 800x600 pixels, toque, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-T12C4D1	PanelView Plus 1250, em cores, toque	Visor TFT de 12 pol., 800x600 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB

Multipainéis SIMATIC - série 37*x*					**Solução Rockwell Automation**		
Código de catálogo Siemens	**Referência rápida**	**Descrição**	**Memória**	**Opções de comunicação**	**Código de catálogo Rockwell Automation**	**Nome**	**Descrição**
6AV6 644-0BA01-2AX0	SIMATIC MP377, teclado, 12,1 pol.	Visor TFT de 12,1 pol., 65.536 cores, 800x600 pixels, teclado, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-K12C4D1	PanelView Plus 1250 em cores, teclado	Visor TFT de 12,1 pol., 800x600 pixels, cores de 18 bits, EtherNet/IP e RS-232, teclado, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6 644-0AB01-2AX0	SIMATIC MP377, toque, 15 pol.	Visor TFT de 15 pol., 65.536 cores, 1024x768 pixels, toque, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-T15C4D1	PanelView Plus 1500, em cores, toque	Visor TFT de 15 pol., 1024x768 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6 644-0BA01-2AX0	SIMATIC MP377, toque, 19 pol.	Visor TFT de 19 pol., 65.536 cores, 1280x1024 pixels, toque, 24 Vcc apenas	12,5 MB	1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, S5, S7-200, S7-300/400, e controladores de terceiros, porta de impressora disponível.	2711P-T15C4D1	PanelView Plus 1500, em cores, toque	Visor TFT de 15 pol., 1024x768 pixels, cores de 18 bits, EtherNet/IP e RS-232, toque, 24 Vcc, 64 MB flash, recursos de impressão USB
6AV6622-0BA01-0AA0	Software WINCC FLEXIBLE STANDARD	Software de configuração e programação para Simatic OP/TP/ MP270, MP370, OP77, OP/TP170 e micropainéis	não aplicável	não aplicável	9701-VWSTMENE	Software RSView Studio Machine Edition	Software de configuração RSView Studio Machine Edition para desenvolvimento e teste aplicativos HMI no nível de máquina

Observações:

Suporte da Rockwell Automation

A Rockwell Automation fornece informações técnicas sobre a Web para ajudar você a usar seus produtos. Em http://support.rockwellautomation.com, você pode encontrar manuais técnicos, uma base de conhecimentos de perguntas mais freqüentes, notas técnicas e de aplicação, código de amostra e links para pacotes de serviço de software e um recurso MySupport que pode ser personalizado para fazer o melhor uso dessas ferramentas.

Para obter um nível adicional de suporte técnico por telefone quanto à configuração, instalação e resolução de problemas, oferecemos programas de suporte TechConnect. Para obter mais informações, entre em contato com seu distribuidor local ou representante Rockwell Automation ou visite http://support.rockwellautomation.com.

Assistência à instalação

Se ocorrer um problema nas primeiras 24 horas da instalação, analise as informações contidas neste manual. Você pode também entrar em contato com um número de suporte ao cliente para ajuda inicial para começar a usar seu produto.

Estados Unidos	1.440.646.3434 Segunda à sexta-feira, das 8:00 às 17:00 EST
Fora dos Estados Unidos	Entre em contato com seu representante local da Rockwell Automation para questões de suporte técnico.

Retorno de satisfação com um produto novo

A Rockwell Automation testa todos os seus produtos para assegurar que estejam totalmente operacionais ao serem embarcados da fábrica.Entretanto, se seu produto não estiver funcionando e precisar ser devolvido, siga estes procedimentos.

Estados Unidos	Entre em contato com seu distribuidor. Você deve fornecer um número de caso de suporte ao cliente (ligue para o número de telefone acima para obter um) a seu distribuidor para concluir o processo de retorno.
Fora dos Estados Unidos	Entre em contato com seu representante Rockwell Automation para o procedimento de devolução.

www.rockwellautomation.com

Sede Mundial para Soluções de Potência, Controle e Informação

Américas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europa/Oriente Médio/África: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Bruxelas, Bélgica, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Ásia-Pacífico: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Brasil: Rockwell Automation do Brasil Ltda., Rua Comendador Souza, 194-Água Branca, 05037-900, São Paulo, SP, Tel: (55) 11.3618.8800, Fax: (55) 11.3618.8887, www.rockwellautomation.com.br
Portugal: Rockwell Automation, Tagus Park, Edifício Inovação II, n 314, 2784-521 Porto Salvo, Tel.: (351) 21.422.55.00, Fax: (351) 21.422.55.28, www.rockwellautomation.com.pt

Publicação LOGIX-AP008B-PT-P - Junho 2008
Substitui a publicação LOGIX-AP008B-PT-P

Copyright © 2008 Rockwell Automation, Inc. Todos os direitos reservados. Impresso nos EUA